Facultat d'informàtica de Barcelona

Treball final de grau

Cerca de trajectòries de pacients a través de les etapes d'una malaltia a partir d'històries digitals

Albert Serven Graupera

Director

Ricard Gavaldà Mestre

Codirector

Jaume Baixeries Juvillà

20 d'octubre de 2016

Agraïments

M'agradaria agrair als meus directors de projecte Ricard Gavaldà i Jaume Baixeries. Els agraeixo de tot cor la paciència, el suport, els consells i la dedicació proporcionada. La seva implicació en el projecte ha estat clau per a poder entregar un bon treball de final de grau. També vull agrair a la doctora Juliana Ribera per l'interès mostrat pel projecte i l'ajuda prestada.

Índex

1. Resum

1.1 Català

Aquest projecte sorgeix d'una col·laboració entre l'empresa Serveis Mèdics i el grup de recerca LARCA de la UPC. Vol aportar una manera diferent d'analitzar les dades de que disposen als professionals de la medicina. S'han treballat diferents tècniques de mineria de dades per a aquest propòsit i les seleccionades finalment han estat encapsulades dins d'una aplicació.

Es vol que aquesta aplicació sigui utilitzada pels metges i gestors de recursos amb les dades de diagnòstics de malalties de pacients, permetent una forma ràpida d'analitzar aquestes dades que d'altra manera no s'hauria arribat a aconseguir.

L'aplicació permet guardar documents amb les dades dels pacients. Aquestes dades s'utilitzen a l'aplicació per a trobar patrons freqüents en les malalties, generar models predictius, grafs i estadística descriptiva. Finalment quan s'han obtingut els resultats, aquests es poden exportar.

Per acabar es fa un anàlisi d'unes dades proporcionades per Serveis Mèdics per a provar l'aplicació. Es fan servir totes les funcionalitats i s'extreuen conclusions.

1.2 Castellà

Este proyecto surge de una colaboración entre la empresa Serveis Mèdics y el grupo de investigación LARCA de la UPC. Quiere aportar una manera diferente de analizar los datos de que disponen los profesionales de la medicina. Se han trabajado diferentes técnicas de minería de datos para este propósito y las seleccionadas finalmente se han encapsulado en una aplicación.

Se quiere que esta aplicación sea utilizada por médicos y gestores de recursos con los datos de diagnósticos de enfermedades de pacientes, permitiendo una forma rápida de analizar estos datos que de otra forma no se podría conseguir.

La aplicación permite guardar documentos con los datos de los pacientes. Estos datos se utilizan en la aplicación para encontrar patrones frecuentes en las enfermedades, generar modelos predictivos, grafos y estadística descriptiva. Finalmente cuando se han obtenido los resultados, estos se pueden exportar.

Finalmente se hace un análisis de unos datos proporcionados por Serveis Mèdics para probar la aplicación. Se usan todas las funcionalidades y se extraen conclusiones.

1.3 Anglès

This project starts from collaboration between the company Serveis Mèdics and LARCA research group of the UPC. It wants to bring a different way of analyzing the data available to medical professionals. For this purpose a search between different data mining techniques has been done and finally the selected ones have been encapsulated in an application.

With the data of patient's diseases it is wanted to use this application by physicians and resource managers, allowing a quick way to analyze these data that otherwise could not be achieved.

The application allows saving documents with patient data. These data is used in the application to find frequent patterns in diseases, generate predictive models, graphs and descriptive statistics. Finally when the results have been obtained, these can be exported.

Finally, an analysis of data provided by Serveis Mèdics to test the application is made. All features of the application are used and conclusions are drawn.

2. Introducció

2.1 Formulació del problema

El sistema sanitari avui en dia disposa d'uns recursos molt limitats i cal saber invertir-los correctament. La planificació i gestió d'aquests representa una gran responsabilitat.

Ja fa un temps que la majoria d'organitzacions del sistema de la salut enregistren les dades dels diagnòstics, anàlisis i prescripcions dels seus pacients digitalment. Aquestes històries digitals creixen molt ràpidament i els experts mèdics no tenen eines per a poder extreure tota la informació que necessitarien saber a l'hora de planificar i fer diagnòstics. Sí que es fan servir dia a dia però amb tècniques elementals que no aprofiten tota la informació que podrien aportar.

Els estudis amb aprenentatge automàtic són ideals per a resoldre aquest problema. Els estudis amb eines tradicionals d'estadística com ara anàlisi multivaluat o bé regressions lineals, no són suficientment potents ja que quan es tracta de diagnosticar o planificar no depèn d'un o pocs factors, hi ha molts i aquests actuen de maneres molt diferents. El que es vol és veure quin és l'estat que inicia la malaltia, per quines etapes passa el pacient, quins problemes de salut se li van diagnosticant, perquè evolucionen cap a un estat o cap a un altre.

La intenció d'aquest projecte és contribuir a demostrar la potencial utilitat de les tècniques de mineria de dades en històries digitals de pacients. A la llarga, podrien fins i tot servir per establir o modificar guies mèdiques, definir noves polítiques de prevenció o bé proposar estudis clínics. Per a poder resoldre el problema de manera concreta, es disposa de dades de l'empresa Serveis Mèdics[1].

Cal dir que un principi aquest projecte anava a rebre dades de Catsalut, específiques de pacients que han tingut infarts. S'anava a aplicar les tècniques de mineria de dades a aquestes dades però es va rebre la notícia després d'esperar uns mesos, que les dades no arribarien a temps. El motiu era que la signatura del conveni necessari per a transferir les dades s'endarreria degut a canvis importants en la direcció de Catsalut. En comptes de fer l'anàlisi específic aleshores es va optar per fer un software més genèric i es va aconseguir que l'empresa Serveis Mèdics proporcionés unes dades per a l'avaluació de l'aplicació.

2.1.1 Objectius

Dividint el problema en objectius, es poden dividir en els següents:

- Veure quines tècniques de mineria de dades poden ajudar a interpretar les dades.
- Implementació d'algunes d'aquestes tècniques de manera que puguin ser utilitzades per professionals de la medicina.
- Visualització dels resultats de la manera més entenedora possible.

[1] http://www.serveismedics.net/cat

2.2 Abast

Aquest projecte s'emmarca dins d'un conveni de col.laboració entre Serveis Mèdics i el grup de recerca LARCA de la UPC per a investigar el potencial de les tècniques de mineria de dades i aprenentatge automàtic en històries digitals. Es vol dissenyar una aplicació per a poder obtenir informació rellevant pels doctors i planificadors que la faran servir.

El que es vol aconseguir són resultats que permetin al personal tècnic obtenir nou coneixement que finalment ajudi vers la planificació i tractament de les malalties dels pacients. Cal processar les dades de manera que hi hagi uns paràmetres d'entrada que regulin el nivell de confiança dels resultats. També s'ha de poder modificar el resultat o bé poder filtrar per malaltia i relacions entre elles.

Es volen implementar diverses funcionalitats que corresponen a les necessitats expressades per Serveis Mèdics, entre les quals hi ha:

- Afegir pacients.
- Filtrat de pacients per característiques.
- Visualitzar estadístiques sobre els pacients.
- Generar conjunts i seqüències freqüents.
- Generar regles d'associació.
- Avaluar models d'aprenentatge automàtic.
- Visualitzar graf de malalties.
- Guardar els resultats.

2.2.1 Posibles obstacles

Com en tot projecte, poden sortir obstacles durant tot el seu curs. A continuació es llista els que es van considerar més rellevants i es proposa una possible solució.

- Les dades que es proporcionen no s'adapten a l'entrada que el software a programar necessita. De manera que o bé s'ha d'adaptar d'alguna manera corregint les entrades o bé hi ha alguna entrada de totes les que hi ha que no és correcte. La solució a aquest obstacle és fàcil i requereix poc temps, només cal posar controls de qualitat i integritat en el programa.
- Els resultats que s'obtenen no són els esperats. Després de consultar amb els tècnics que faran servir el software es troba que els resultats obtinguts no s'aproximen a la realitat. Per exemple, els algorismes de predicció tenen una capacitat predictiva molt baixa. La solució d'aquest obstacle pot ser o bé ràpida o molt lenta depenent de l'escenari.
- L'eina de visualització no és fàcil de fer servir per part dels tècnics o necessita un aprenentatge massa llarg pel seu temps o paciència. Es pot solucionar fent un estudi d'usabilitat amb usuaris reals per millorar l'experiència d'usuari.

2.3 Metodologia i rigor

Existeix una gran varietat de metodologies pel desenvolupament de projectes de software. L'existència d'aquestes és degut a les diferents situacions en les que un projecte pot realitzar-se. En aquest cas, el projecte es farà en un període curt de temps, uns sis mesos. Per aquest motiu, els cicles de producció i control s'han d'adaptar i reproduir-se cada poc temps.

Les metodologies més apropiades per aquesta mena de casos s'anomenen metodologies àgils. Es basen en el desenvolupament iteratiu i incremental, en el que els requisits i solucions evolucionen a mesura que avança el projecte. Això permet fer correccions abans de que el projecti agafi un camí equivocat.

Per a aquest projecte, s'ha triat una d'aquestes metodologies àgils, anomenada Scrum.

2.3.1 Metodologia Scrum

Aquesta metodologia és un procés en el que s'apliquen de manera regular un conjunt de bones pràctiques per treballar en equip i obtenir el millor resultat possible d'un projecte. Usant Scrum el projecte s'executa en blocs temporals curts i fixos. Cada iteració ha de proporcionar un resultat complet, un increment en el producte final que sigui susceptible de ser entregat. Les activitats que es duen a terme són les següents:

- **Planificació de l'iteració:** aquesta és la primera part de l'iteració que es porta a terme. Té com objectius principals seleccionar els requisits de tota l'iteració i elaborar una llista de tasques que s'han de fer.
- **Execució de l'iteració:** cada dia es comença amb una reunió en la que tots els membres de l'equip inspeccionen el treball que s'ha realitzat el dia anterior. Aquesta reunió no pot durar més de 15 minuts i s'ha de respondre a les preguntes: Que he fet des de la última reunió de sincronització? Que faré a partir d'aquest moment? Quins obstacles tinc o tindré?

  Al respondre aquestes preguntes, l'equip continua amb el treball que se li hagi assignat. En el cas d'aquest projecte i per motius d'horaris entre director i alumne, les reunions seràn més del caire setmanal amb el suport de videoconferències i correus electrònics.
- **Inspecció i adaptació:** L'últim dia de l'iteració es realitza la reunió de revisió de l'iteració. Té dues parts:
  - *Demostració*: es fa una demostració del que s'ha fet en aquella iteració. En funció dels resultats obtinguts i dels canvis que s'hagin fet en el context del projecte, es realitzaràn les adaptacions necessàries de manera objectiva, replanificant el projecte.
  - *Retrospectiva*: l'equip analitza com ha estat la seva manera de treballar i quins són els problemes que podríen impedir-li progressar adequadament, millorant de manera contínua la seva productivitat.

2.3.2 Mètodes de treball

Cada dilluns es marcaràn tasques que hauràn de ser realitzades en el transcurs de la setmana. Per a un major control de les mateixes hauràn de ser estimades en menys de 3h de feina. Per a realitzar aquest control, s'utilitzaràn eines com ara software especialitzat per a aquest tipus de tasques (*Trello*, *Podio*, *Wrike*).

Es farà servir una eina de control de versions amb repositori de codi (*Github*, *Bitbucket*), això permetrà una millor organització i seguretat. Per una banda, cada nova funcionalitat a implementar es dividirà en branques amb el fi de dinamitzar les implementacions i asegurar el funcionament correcte de la resta del software. Per una altra banda, s'assegura que no es perden hores de feina per un error humà relacionat amb els fitxers, com ara borrar parts de codi o arxius accidentalment.

Al acabar la setmana es durà a terme una evaluació de les tasques realitzades i en cas de no haver arribat al rendiment esperat, es prendran les mesures necessàries per a la següent setmana.

2.3.3 Eines de seguiment

Quan una consultoria realitza un projecte amb metodologies àgils és molt habitual quedar amb el client cada setmana o cada dues com a molt. Com ja s'ha comentat abans, aquest comportament es pot aplicar a aquest projecte i reunir-se amb el director cada setmana per informar de les tasques realitzades i resultats obtinguts a part de planificar la següent setmana o bé haver d'adaptar els plans si s'escau.

2.3.4 Mètodes de validació

Per a cada tasca, a més a més de la seva descripció, s'afegirà un apartat on expliqui quan una tasca està acabada i així poder limitar el seu abast. Per a les tasques relacionades amb el desenvolupament de codi s'afegirà una serie de tests. D'aquesta forma assegurem el correcte funcionament del software, aquesta metodologia és coneguda com Test Driven Development (TDD).

Test Driven Development

TDD és un procés de desenvolupament de software basat en la repetició d'un cicle de desenvolupament molt curt. Pot definir-se com el cumpliment d'aquests tres passos:

- El desenvolupador escriu un test automàtic que defineix la millora o nova funcionalitat
- Es produeix la quantitat mínima de codi per a passar el test
- Es realitza una restructuració del codi per a que estigui dins dels estàndars de desenvolupament del projecte

2.4 Context

Actualment, la sanitat catalana està recopilant gran quantitat d'informació sobre els seus pacients, però no s'extreu tot el coneixement que conté. Aquestes dades realment són molt útils per a millorar la planificació i el tracte que reben els pacients, encara que els experts no tenen com analitzar correctament i amb certesa aquesta informació degut a la gran quantitat d'històries digitals de que es disposa.

Degut a que el sistema sanitari és una de les grans despeses de l'administració pública cal administrar bé els recursos que li cedeixen i per tant qualsevol millora possible envers l'estalvi de recursos o bé millora d'atenció als pacients, que finalment repercutirà a una despesa menor, és necessària. L'aplicació de tècniques de mineria de dades a l'informació digital que s'està recopilant pot afavorir a extreure el coneixement necessari que fa falta per a millorar el sistema que de cap altra manera seria possible.

En aquest projecte es disposarà de dades sobre diagnòstics de pacients proporcionades pel Serveis Mèdics. El que es busca per part del personal mèdic és poder treure informació útil com ara patrons en les malalties d'un pacient. Amb aquest projecte es vol aconseguir trobar aquesta informació dels pacients a partir de les dades proporcionades i mostrar-les de manera entenedora.

2.4.1 Actors implicats

El projecte estarà dirigit per en Ricard Gavaldà i Jaume Baixeries, professors a la Facultat d'Informàtica de Barcelona (FIB). La proveïdora de les dades ha estat la doctora Juliana Ribera per part de Serveis Mèdics.

Els beneficiaris directes del software seran els doctors i planificadors que el faran servir. Els ajudarà a fer un anàlisi més complet i a prendre mesures en conseqüència als resultats que s'obtinguin. Donat que hi haurà decisions més encertades, els pacients podran tenir un millor tractament mèdic i per tant beneficiar-se també del projecte.

2.5 Estat de l'art

El fet d'utilitzar tècniques de mineria de dades i aprenentatge automàtic al camp de la medicina no és nou, utilitzant les històries digitals, anomenades internacionalment com a Electronic Health Records (EHR) per a extreure informació.

Aquests estudis es poden classificar en dos tipus, els que es concentren en alguna malaltia o grup de malalties concretes i els que són genèrics. Per veure on està la frontera del coneixement i així resumir una mica el que s'ha fet en aquest camp, a continuació s'expliquen els estudis que s'han realitzat.

2.5.1 Estudis específics

Hi ha una gran quantitat d'estudis on s'han aplicat tècniques de mineria de dades a malalties concretes o grups de malalties. Aquests estudis busquen un nivell de detall molt gran i extreuen coneixement de l'informació que només és aplicable a aquella malaltia.

El cas de l'estudi [MFC13] s'aplica un anàlisi enriquit per a detectar característiques en els pacients que siguin diferenciadores en les possibles etapes de la malaltia. La malaltia en concret és la miocardiopatia hipertròfica. El que es vol aconseguir és predir si el pacient es troba en una fase o una altra i així poder prendre mesures significatives al respecte.

L'estudi [YL06] troba malalties relacionades amb la hèrnia paraesofàgica mitjançant regles d'associació a partir de les malalties que pateixen els pacients amb la hèrnia. Aquests resultats es validen amb els experts en medicina.

2.5.2 Estudis genèrics

En aquests estudis genèrics es busca fer un aprenentatge no supervisat que pugui trobar nou coneixement mèdic. De manera que es vol obtenir patrons o relacions ocurrents que no s'han tingut en compte i d'aquesta manera obrir noves vies d'hipòtesis on els especialistes mèdics poden recolzar-se.

En el cas de l'estudi [MSL+06], utilitza diferents metodologies de mineria de dades per a trobar noves associacions entre malalties a partir dels EHR de més de mig milió de pacients.

En un altre estudi, [HRC09], s'usen també EHR per a crear xarxes de malalties i agrupar-les per aparició en un sol pacient. És a dir, tenen una relació directa perquè es detecta que un pacient té símptomes de les malalties que s'han agrupat.

Altres estudis busquen associacions de parells de malalties en pacients mitjançant heurístiques i una sèrie d'estadística que acaba filtrant les més rellevants, és el cas de l'estudi [CMM+05]. A l'estudi [GCV+07], es crea un graf de malalties relacionades entre sí i a la vegada s'intenta relacionar amb problemes de salut documentats.

Un altre estudi genèric, similar al que es vol tractar, [MTIV97], que presentaven uns anàlisis descoberts a partir dels patrons de progressió de malalties fent servir també EHR que cobrien tota la població de Dinamarca. Donades les dades, trobaven les trajectòries

més significants i les agrupaven en grups de malalties relacionades amb l'obstrucció pulmonar crònica i la gota. El que volien era mitigar el risc i predir/prevenir futures malalties pels pacients.

Finalment cal comentar dos treballs de final de grau[2], dins del mateix grup de recerca, un extensió de l'altre. Els estudiants són Martí Zamora i Manel Baradad. En ambdós casos, s'agafaven dades de pacients proporcionats per l'Institut Català de la Salut, amb una estructura de projecte similar a la d'aquest, a partir d'elles es relacionava malalties i medicaments per tal de veure quins medicaments corresponien a una malaltia. També es produïen alarmes quan es detectaven receptes de medicaments que no són habituals amb una certa malaltia. Finalment es visualitzaven els resultats en un graf.

2.5.3 Anàlisi

Observant els estudis previs i les tècniques emprades en cada un el projecte que es planteja necessita fer servir un algorisme de mineria del que s'anomena patrons freqüents. Es vol aconseguir una flexibilitat en l'entrada de dades, fent que amb poc esforç qualsevol entitat pugui fer servir les seves. S'ha d'analitzar els algorismes que existeixen i veure quins poden resultar útils per aquest projecte.

A Catalunya, aquest tipus d'estudi en concret creiem que no s'ha realitzat. Els dos treballs de final de grau abans esmentats trobaven conjunts de malalties freqüents però sense tenir en compte el seu ordre en el temps, és a dir, no calculaven trajectòries. En aquest sentit, el resultat dels estudis que es vol permetre fer s'assembla més a [JMO+14].

[2]https://upcommons.upc.edu/handle/2099.1/23152 i https://upcommons.upc.edu/handle/2117/78672

3. Definició del projecte

Aquest projecte proposa tècniques de mineria de dades i estadística descriptiva perquè faciliti als metges i gestors de recursos mèdics la comprensió de les dades que disposen.

Donat unes dades inicials amb les corresponents malalties i pacients, l'eina permet les funcionalitats següents:

- Llegir les dades en el format proporcionat per Serveis Mèdics.
- Veure estadístiques dels atributs de les dades, com ara edat, gènere i visites al metge.
- Calcular els conjunts de malalties més freqüents que apareixen simultàniament en la població.
- Donada una data en el temps de quan es va registrar la malaltia, mostra les seqüències més freqüents de malalties en el temps, les anomenades "trajectòries freqüents".
- Formulació d'un graf que relaciona malalties.
- Veure regles d'associació entre malalties.
- Exportar les dades dels resultats.

Aquestes funcionalitats s'encapsulen en una aplicació amb una interfície gràfica que ajuda a fer-les servir.

El projecte primer de tot ha buscat quines solucions existeixen actualment en aquest àmbit, les tècniques de mineria de dades més adients per a fer servir i es va fer un primer esbòs de l'aplicació, a nivell gràfic.

Un cop fetes totes les decisions es passa a l'implementació del programa. Es pot dividir l'implementació en desenvolupament de les funcionalitats, interfície gràfica i finalment ajuntar aquestes dues parts per a que funcionin com una de sola.

Un cop implementat, amb les dades proporcionades de Serveis Mèdics s'ha comprovat el correcte funcionament de l'aplicació i a la vegada s'han analitzat els resultats obtinguts.

4. Planificació

A continuació s'expliquen les tasques que conformaran el projecte, així com el temps i recursos requerits d'aquestes juntament amb el raonament de la seva seqüència lògica i les dependències de precedència. Es proposen possibles solucions a incompliments amb la programació estipulada o com afectarien en la durada del projecte.

4.1 Descripció de les tasques

El projecte està dividit en tres parts essencials: la fita inicial, el projecte i la fita final. A la fita inicial es fa la gestió del projecte, la part del projecte conté tota la part tècnica i pràctica, i finalment la fita final on s'entrega la memòria i es fa la presentació. A continuació es fa la descripció i una estimació del temps requerits per cada tasca.

Els recursos que calen per a aquest projecte són pocs, al ser una eina de programari només cal un ordinador amb el sistema operatiu Unix i connexió a internet per a descarregar el que faci falta. En el meu cas, faig servir un Macbook Pro de 15 polsades amb una distribució de Linux anomenada Linux Mint (actualment amb versió 17).

A nivell de recursos humans, s'ha plantejat tenir un director i codirector de projecte, un *data scientist* per a fer mineria de dades, aprenentatge automàtic i l'experimentació amb les dades proporcionades un desenvolupador web per a l'interfície del programa, la comunicació amb les crides a les tècniques de mineria de dades i entrada/sortida de dades orientades en un entorn web.

4.1.1 Fita inicial

En aquesta part es planteja la gestió del projecte i conté les següents tasques:

- **Abast del projecte i contextualització** (25h.): Definir l'objectiu del projecte, com es desenvoluparà i amb quins mitjans. També el perquè de la temàtica i la contextualització.
- **Planificació temporal** (8h.): Es planifica el projecte en el temps, s'especifiquen totes les tasques que hi haurà, els requeriments i els recursos associats.
- **Gestió econòmica i sostenibilitat** (9h.): Càlcul del pressupost del projecte i valoració de la seva sostenibilitat.
- **Presentació preliminar** (6h): Fer una presentació amb els materials generats de les tasques anteriors.
- **Presentació oral i document final** (18h.): Síntesi d'aquesta fase inicial que conté totes les tasques anteriors més una justificació de les competències tècniques del projecte i adequació a l'especialitat.

4.1.2 Projecte

El treball final de grau en si, és el projecte on hi haurà la part tècnica. Com la metodologia que s'ha triat va guiada per iteracions (*Scrum*), la part principal del projecte constarà de tasques que seran cada una d'aquestes iteracions. És a dir, al llarg de la planificació s'aniran fent tasques per implementar funcionalitats, al acabar una comença la següent, totes amb una estimació de temps similar, encara que hi haurà que requeriran més temps i d'altres menys. Finalment caldrà una tasca que agafi totes aquestes funcionalitats i generi l'aplicació final.

S'ha decidit separar l'aplicació en dues parts diferenciades: la part d'interacció amb les dades i visualització i la que fa els càlculs, anomenats normalment com client i servidor respectivament. La proposta de tasques a realitzar és la següent:

- **Preparació de l'entorn de treball** (8h.): Aquesta tasca consistirà en instal·lar l'entorn d'eines que es faran servir per a dur a terme el projecte. Caldrà un ordinador amb el sistema operatiu Linux i les eines necessàries per a poder programar: editor, llenguatge/s de programació, llibreries i altres programes a tenir en compte. Cal provar que tot funcioni correctament.
- **Anàlisi i tria de mètodes a usar** (60h.): Cal veure quins mètodes són factibles per a aplicar a aquest projecte de manera que el resultats obtinguts siguin satisfactoris. Quines tècniques i algorismes de mineria de dades caldria fer servir, les estructures de dades que serviran per a guardar els resultats i quines eines de visualització seran millors per a mostrar aquests resultats. En aquesta tasca també caldrà veure l'organització interna del programa (arquitectura) i considerar les diferents possibilitats d'implementar a recursos externs (via API, sense recursos, bases de dades).
- **Disseny de l'aplicació** (20h.): Caldrà tenir un disseny gràfic del programa, interaccions possibles que es podran fer, quines vistes tindrà, l'accessibilitat i altres decisions que tenen a veure amb l'experiència d'usuari.
- **Implementació de l'aplicació** (135h.): Durant aquest període cal aplicar les decisions del punt anterior, és a dir, implementar els mètodes i disseny seleccionats. Cal automatitzar l'entrada de dades, crear una interfície per a poder realitzar els experiments amb el programa i la visualització. En aquesta tasca s'inclou una mica de proves per a veure que l'implementació sembla que es comporta com és esperat. En la següent tasca és quan realment es comprova que les solucions són vàlides. Al ser una tasca que dedica moltes hores, es pot dividir en subtasques:
  - **Entrada i sortida de dades** (20h.): El programa a de ser capaç de rebre unes dades, en un format especificat i guardar els resultats que s'obtenen. Aquesta part d'implementació cal tenir en compte possibles inconsistències en les dades, de manera que el programa no falli en aquests casos.
  - **Implementació de les tècniques de mineria de dades** (50h.): Cal veure que les llibreries de mineria de dades fetes servir reben correctament les dades, els algorismes que s'han triat cal que proporcioni resultats amb les dades que se li passen. Cal també adaptar la comunicació amb l'aplicació.

  - **Interfície gràfica i visualització (client)** (45h.): La interfície del programa vindrà preestablerta amb un disseny que s'haurà pensat en la fase anterior com les llibreries per a la implementació. Aquesta part haurà de comunicar-se amb el servidor, on hi haurà la part de càlcul.
  - **Servidor** (20h.): El servidor rebrà les dades que li arribin del client i cridarà als mètodes de mineria de dades implementats, retornant els resultats al client que s'encarregarà de gestionar-los.

- **Experimentació amb dades proporcionades** (150h.): En aquesta tasca caldrà provar que l'aplicació que s'ha implementat de manera correcta, que totes les funcionalitats es comporten com s'espera i no hi ha errors no previstos. Es farà un anàlisi de les dades usant l'aplicació, caldrà un petit estudi previ de l'estructura de les dades, ajustaments per a poder entrar les dades al programa.

Cal dir que entre tasques i durant la realització de les mateixes hi haurà reunions amb el director del projecte per veure que tot té un curs correcte i si no seguís aquest curs, poder corregir les desviacions. Aquestes reunions tindran una durada aproximada d'una hora on s'explicarà el que s'ha estat treballant, les tasques que s'han completat i el treball que es farà per la pròxima setmana.

4.1.3 Fita final

Un cop s'ha fet el projecte, el que cal és redactar la memòria i preparar la presentació final. Aquesta memòria contindrà la part escrita en la fita inicial més el que s'ha treballat en el projecte: aspectes tècnics com ara implementacions, decisions, mètodes d'anàlisi, especificació detallada de les característiques del programa generat i d'altres.

Pel que fa la presentació final, seria un resum de tot el projecte capaç d'explicar tot el que s'ha fet de manera entenedora i en poc temps. S'han de fer servir taules, esquemes i imatges rellevants per acompanyar la presentació i ajudar a la comprensió. El total d'hores dedicades aproximadament en aquestes tasques hauria de ser sobre les 30, dues setmanes dedicant 3 hores al dia de dilluns a divendres.

4.1.4 Seqüència lògica i dependències

Les tasques descrites per aquest projecte poden ser fàcilment paral·lelitzables però al tractar-se d'un projecte unipersonal s'ha de fer seqüencialment. Les tasques de la fita inicial tenen dependències en l'ordre de menció però mentre es fan aquestes tasques es pot mirar de fer la tasca de preparació de l'entorn de treball, el preprocessament de les dades i l'anàlisi i tria de mètodes a utilitzar.

A l'hora d'implementar el programa i validar el resultats, es pot organitzar de manera que parts de la implementació poden ser provades i validades a mesura que s'avança en el projecte sense haver de tenir tot el programa complet i funcionant. La fita final es pot anar fent a mesura que es fa el projecte, documentar poc a poc a mesura que es van implementant és possible i així es repartiria la tasca de fer la memòria en subtasques que consistirien en fer parts d'aquesta. Ara bé, la seqüència lògica seria seguir les tasques com s'han enunciat en aquest document.

4.2 Valoració d'alternatives i pla d'acció

4.2.1 Possibles desviacions temporals

En el projecte pot passar que una tasca porti més temps de l'estimat a completar-la. Per això cal tenir en compte possibles endarreriments de la planificació per tal de re-adreçar totes les tasques restants i seguir així complint el temps estipulat.

Problemes d'instal·lació

Un problema habitual que acostuma a endarrerir un projecte seria trobar-se amb errors al instal·lar un programa. Ja siguin per qüestions d'incompatibilitat, falta de programari requerit o versió de sistema operatiu, aquest problema endarrereix tot el projecte perquè es precisa de l'instal·lació correcta el poder començar. Per tant és molt important solucionar-ho quant abans millor ja que la repercussió en la planificació es directa.

El que cal és simplement dedicar més hores fins a solucionar-ho o pot ser buscar una alternativa equivalent que serveixi i no doni problemes. Aquí el consum de recursos només són les hores i normalment és una desviació petita.

Dificultats d'implementació

El que hauria de ser una implementació estimada en el un determinat temps, augmenta l'estimació en més hores. En aquest cas caldria fer una reunió amb el director del projecte i avaluar si es possible finalitzar l'implementació en un temps considerable extra o bé veure quines alternatives al que s'està intentant realitzar es poden aplicar. Com ara, l'algorisme triat és massa complicat per a poder entendre'l i fer que funcioni correctament, el que es podria fer és simplificar-lo.

Errors en les proves

Si es trobessin errors en les proves avaluant programa o els especialistes mèdics veuen que no concorden els resultats amb el que s'espera, solucionar-ho dedicant-hi més hores. Depenent del tipus d'error que es cometi l'impacte a la planificació serà inapreciable o bé catastròfic. Per aquest motiu cal tenir un seguiment d'aquestes proves i resultats cada cert temps.

4.2.2 Finalització del projecte

Donada la planificació de les tasques i valorades les possibles desviacions temporals, el projecte s'assegura acabar al finalitzar el quadrimestre, amb la metodologia àgil que s'ha triat, les possibles desviacions previstes o imprevistes es poden corregir ràpidament i d'aquesta manera s'assegura que el projecte es pot finalitzar a temps i sense problemes.

4.3 Diagrama de Gantt

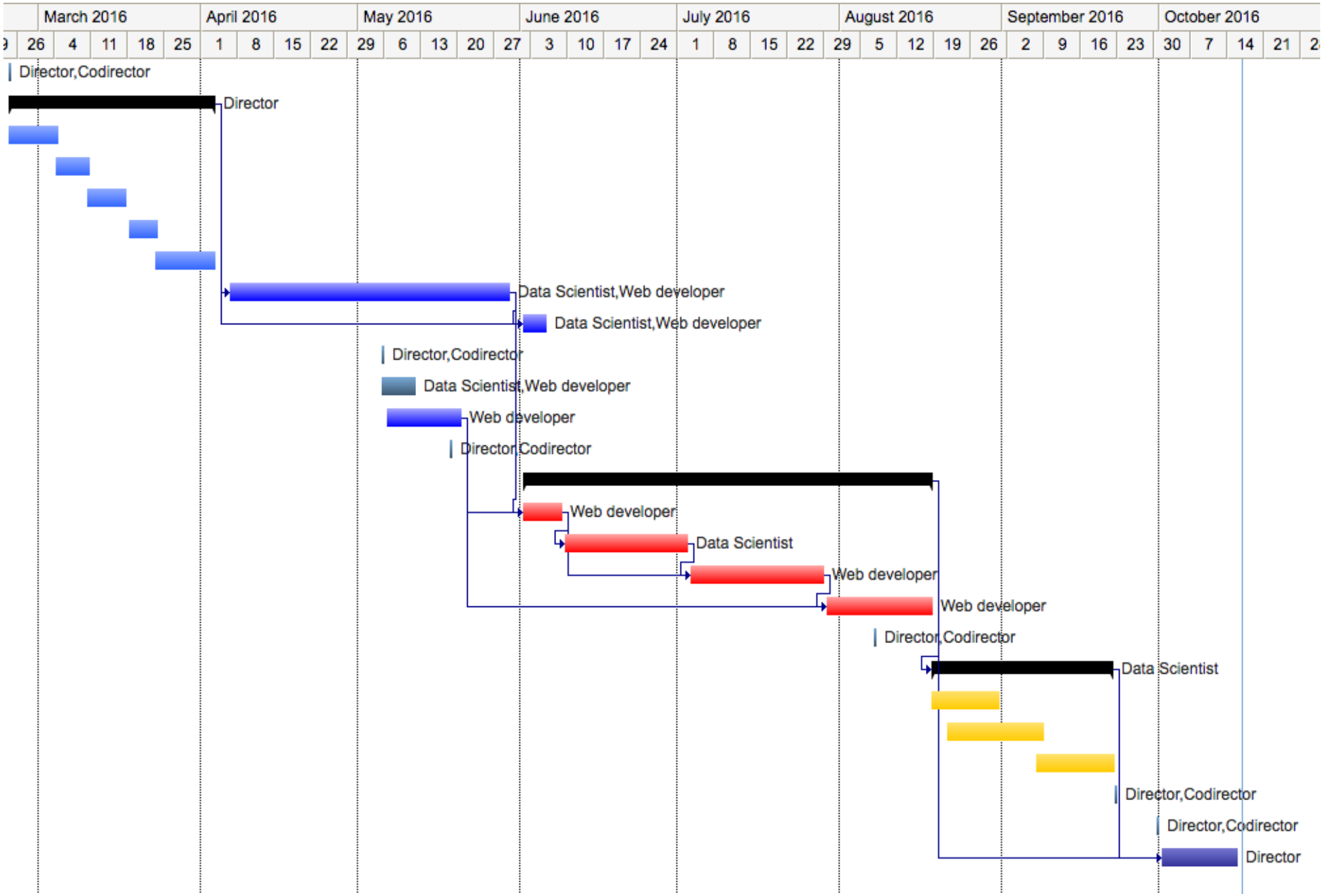
Donades les tasques abans descrites, en les següents figures es mostren la seva organització en el temps, els rols que durà a terme cadascuna i hi apareix un codi de colors segons el nivell de risc: blau per a tasques amb un risc baix, taronja amb les tasques de risc mitjà i finalment vermell per a les tasques de risc alt.

Name	Duration	Start	Finish	Predec	Resources
Primera reunió	1d?	02/22/2016	02/22/2016		Director,Codirector
⊟GEP	**30d?**	**02/22/2016**	**04/01/2016**		**Director**
Abast del projecte i contextualització	8d?	02/22/2016	03/02/2016		
Planificació temportal	5d?	03/02/2016	03/08/2016		
Gestió econòmica i sostenibilitat	6d?	03/08/2016	03/15/2016		
Presentació preliminar	4d?	03/16/2016	03/21/2016		
Presentació oral i document final	10d?	03/21/2016	04/01/2016		
Anàlisi i tria de mètodes a usar	40d?	04/04/2016	05/27/2016	2	Data Scientist,Web developer
Preparació de l'entorn de treball	5d?	05/30/2016	06/03/2016	2,8	Data Scientist,Web developer
Segona reunió	1d?	05/03/2016	05/03/2016		Director,Codirector
Reunió tècnics	5d?	05/03/2016	05/09/2016		Data Scientist,Web developer
Disseny de l'aplicació	10.5d?	05/04/2016	05/18/2016		Web developer
Reunió de seguiment	1d?	05/16/2016	05/16/2016		Director,Codirector
⊟Implementació del programa	**56.5d?**	**05/30/2016**	**08/16/2016**		
Entrada i sortida de dades	6d?	05/30/2016	06/06/2016	8,12	Web developer
Implementació de les tècniques de mineria de dades	18d?	06/07/2016	06/30/2016	15	Data Scientist
Interfície i visualització (client)	18d?	07/01/2016	07/26/2016	15,16	Web developer
Servidor	14.5d	07/27/2016	08/16/2016	12,17	Web developer
Reunió de seguiment	1d?	08/05/2016	08/05/2016		Director,Codirector
⊟Experimentació amb les dades proporcionades	**24.63d?**	**08/16/2016**	**09/20/2016**	**14**	**Data Scientist**
Estudi de les dades	9d?	08/16/2016	08/29/2016		
Anàlisi de les dades	13d?	08/19/2016	09/06/2016		
Comprovació del bon funcionament	11.13d?	09/05/2016	09/20/2016		
Reunió de seguiment	1d?	09/20/2016	09/20/2016		Director,Codirector
Reunió final	1d?	09/28/2016	09/28/2016		Director,Codirector
Fita final	11d?	09/29/2016	10/13/2016	14,20	Director

La tria i anàlisi de mètodes es farà el primer de tot, cal veure quins algorismes fer servir, si són útils per l'aplicació que es vol, les tecnologies que es faran servir tant en client com en servidor.

Les implementacions del software aniran depenent una darrera l'altra per a poder desenvolupar cada funció tenint l'anterior, és a dir, l'algorisme disposarà de les dades ja ben processades i disponibles, i l'interfície ja tindrà els resultats finals que haurà calculat el servidor.

Finalment les proves del programa en general es podran fer simultàniament, encara que es distribueixin una mica en seqüència.


March 2016
April 2016
May 2016
June 2016
July 2016
August 2016
September 2016
October 2016
26 4 11 18 25 1 8 15 22 29 6 13 20 27 3 10 17 24 1 8 15 22 29 5 12 19 26 2 9 16 23 30 7 14 21
Director,Codirector
Director
Data Scientist,Web developer
Data Scientist,Web developer
Director,Codirector
Data Scientist,Web developer
Web developer
Director,Codirector
Web developer
Data Scientist
Web developer
Web developer
Director,Codirector
Data Scientist
Director,Codirector
Director,Codirector
Director

4.4 Identificació i estimació dels costos

Els costos d'aquest projecte es basaràn en els recursos humans, el hardware fet servir, les despeses generals i els impostos. Tot el software que es farà servir és de programari lliure/codi obert i per tant els costos d'aquests no tenen impacte en el projecte.

Recursos humans

Aquí es té en compte els recursos humans amb un salari aproximat a un professional que treballa en una empresa. Farà falta un director i codirector de projecte, un *Data Scientist* per a l'anàlisi de les dades i implementació de les tècniques de mineria de dades. Cal una persona especialitzada en tecnologies web, tant en client per a visualitzar les dades com en servidor per a fer els càlculs, és a dir un *full stack web developer*. A continuació es mostra una taula amb l'estimació dels salaris per a recursos humans:

Rol	**Salari**
Director de projecte (D)	23,45 €/h
Codirector de projecte (C)	23,45 €/h
Data scientist (DS)	20,83 €/h
Full stack web developer(FSD)	18,22 €/h

Taula 4.1: Salaris per a recursos humans

Els salaris s'han basat en els estudis de remuneració de l'any 2016 en la secció de tecnologia de Page Personnel, suposant que el nivell d'experiència de cada persona és alt (més de 3 anys). A continuació, a nivell de les activitats descrites en el diagrama de Gantt, es fa una estimació dels costos d'aquestes activitats i s'especifica qui les durà a terme:

	Hores	Equip implicat	Cost
GEP	75	D, C	3.517,50 €
Anàlisi i tria de mètodes	60	DS, FSD	2.343,00 €
Disseny de l'aplicació	20	FSD	364,40 €
Implementació E/S	20	FSD	364,40 €
Implementació mineria de dades	50	DS	1.041,50 €
Implementació client	45	FSD	819,90 €
Implementació servidor	20	FSD	364,40 €
Experimentació amb dades	150	DS	3.124,50 €
Fita final	30	D, C	1.407,00 €
Reunions setmanals	24	D, C	1.125,60 €
Total			**14.472,20 €**

Taula 4.2: Estimació de cost per activitat

Hardware

Pel que fa el hardware, és farà servir un sol portàtil per a poder desenvolupar el software. S'ha optat per un Macbook donades les comoditats d'un sistema basat en Unix i per la cobertura que proporciona l'empresa Apple. A continuació es desglossa el preu d'aquesta màquina, amb el dóna el preu del hardware en aquest projecte si es fa servir durant uns 6 mesos unes 6 hores al dia (179,28 €).

Cost total	Vida útil	Dies laborables a l'any	Hores/día	Cost/hora
1.500,00 €	4 anys	251	8	0,18675 €/h

Taula 4.3: Estimació cost hardware

Software

El pressupost que es destina a software és nul, perquè tot el programari que s'utilitzarà te llicència gratuïta. Això es deu a que són de programari lliure/codi obert.

Costs indirectes

Com que tota la feina es farà dintre d'institucions públiques, com que funcionen independentment del projecte que es farà a les seves instal·lacions, ni l'electricitat ni el manteniment s'han contemplat com a càrrec en costos indirectes.

Cost total

Concepte	**Cost**
Recursos humans	14.472,20 €
Hardware	179,28 €
Software	0,00 €
Contingència (20%)	
Total	**17.366,64 €**

Taula 4.4: Salaris per a recursos humans

Per aquest projecte s'ha triat un contingència del 20%, per tal de cobrir possibles retards en les tasques com poden ser les d'implementació, o bé reunions inesperades fora de la planificació estipulada.

Control de gestió

Donat que no sempre es pot seguir la planificació establerta o consumir just el que el pressupost ha calculat, cal determinar uns mecanismes per a controlar aquestes desviacions.

Les desviacions menys probables són amb el hardware, ja que es farà servir un ordinador per a desenvolupar el projecte, hi ha poques probabilitats que surgeixi una desviació degut a un possible malfuncionament. Per altra banda, El programari fet servir serà poc probable de causar desviacions donada la gran varietat de possibles solucions en el món del programari lliure/codi obert.

Donada la taula de Gantt amb la descripció de les tasques del projecte, es pot fer un control de desviacions a nivell d'hores calculades i les hores reals realitzades. Es mantindrà un registre de les hores de feina realitzades amb data i concepte per tal de veure si es sobrepassa el número d'hores planificat.

Per a fer el càlcul d'aquestes desviacions es faràn servir aquestes fòrmules:

- Desviament de mà d'obra en preu = (cost est. - cost real) · consum d'hores real
- Desviament en la realització d'una tasca en preu = (cost est. - cost real) · consum d'hores real

- Desviament d'un recurs en preu = (cost est. - cost real) · consum real
- Desviament total en la realització de les tasques = cost est. total tasques - cost real total tasques
- Desviament total en recursos = cost total est. recursos - cost total real recursos
- Desviament total en costos fixes = cost total pressupostat - cost total real

4.5 Sostenibilitat

A continuació es presenta l'estudi sobre sostenibilitat que s'ha fet sobre el projecte. Per a poder valorar en els tres aspectes, econòmicament, socialment i ambientalment s'han seguit una sèrie de pautes que ajuden a guiar la puntuació que ha d'obtenir cada part

4.5.1 Econòmica

Per a veure si aquest projecte es viable econòmicament hi ha una avaluació de costos previs al projecte amb els de recursos humans i es tenen en compte les desviacions possibles que pot tenir el projecte. Degut a la metodologia àgil que es fa servir els ajustaments que es poden fer durant el projecte estan garantits. Si aquest projecte hagués de ser competitiu el pressupost calculat és adient i seria perfectament viable, a més que es podria reutilitzar en altres empreses del mateix sector de la salut amb les seves dades.

Les estimacions a nivell de temps i recursos són ajustades i difícilment es podria millorar el cost en aquest aspecte si es reduïssin les variables anteriors. Es podria fer un projecte similar però segurament no tindria les prestacions a nivell de programa i la solidesa d'aquest. Hi ha una bona repartició de les tasques i s'aprofiten totes les eines disponibles per a fer els càlculs i l'implementació del software.

Donat que és un projecte on participa Serveis Mèdics, és factible que si surt correctament es pugui estendre a altres entitats mèdiques i/o estendre la seva funcionalitat. Donat que té molts punts a favor econòmicament la qualificació que se li pot atorgar és d'un 9.

4.5.2 Social

El projecte es realitza en col·laboració amb Serveis Mèdics. Actualment el sector sanitari està molt castigat per les retallades en el seu pressupost i les col·laboracions d'aquesta mena són poc freqüents i es necessita de bons resultats que afavoreixin a una millora de la sanitat en general per a què se'ls hi doni més importància.

Aquest projecte pot afavorir molt la situació en que es troba la sanitat pública si es fa servir correctament, tant a nivell de pacients, metges i com econòmic, ja que comporta gastar menys en medicació si es pot veure les trajectòries de la malaltia del pacient abans i més còmodament per part del especialista mèdic.

També cal dir que no hi haurà cap col·lectiu que pugui sorgir perjudicat, beneficia a tant els usuaris del programa (metges), tant en els hospitals i institucions amb la millora de presa de decisions i finalment els pacients que rebran un millor tractament.

En aquest aspecte el projecte resulta molt beneficiós i la qualificació que se li posaria és de 10.

4.5.3 Ambiental

Aquest projecte necessitarà d'un ordinador personal per a poder desenvolupar-se i res més. Al ser un projecte de creació de programari el consum i l'impacte ambiental que deixarà és mínim i negligible ja que és farà servir dins un ordinador amb molts altres programes funcionant.

Com que tot el projecte es realitza via online, tant la documentació i el programa, els recursos materials que consumirà seran zero. L'aprofitament d'altres eines de programari és benefici ja que s'estalvien temps i diners en implementacions. Tampoc cal preocupar-se de contaminació i reciclatge.

Degut a que el seu impacte és nul, aquest apartat també te una qualificació de 10.

4.5.4 Matriu de sostenibilitat

La matriu de sostenibilitat que resulta finalment és:

	Econòmica	Social	Ambiental	
Projecte Posat en Producció	Viabilitat econòmica 9/10	Millora de la qualitat de vida 10/10	Anàlisi de recursos 10/10	**29/30**
Vida útil i resultats	Cost final 17/20	Impacte social 18/20	Petjada ecològica 19/20	**54/60**
Riscs	Riscs econòmics -10	Perjudicis socials -5	Perjudicis ambientals -2	**-17**
Total	**66/90**			

Taula 4.5: Matriu de sostenibilitat amb les qualificacions

5. Patrons freqüents i aprenentatge automàtic

Una de les principals funcionalitats que presenta aquest projecte és la cerca de patrons freqüents, a continuació s'explica en què consisteix i els principals algorismes que apliquen aquestes cerques. Donat que l'aplicació no implementa cap de les tècniques i algorismes, simplement les utilitza com una caixa negra, les descripcions no aniran al detall de cadascun, simplement els trets característics.

Una manera de trobar patrons freqüents és comptar el nombre de vegades que un conjunt d'elements apareix en unes dades i fixant-se en els que apareixen més sovint. Una altra forma pot ser agafant seqüències d'aquests conjunts classificant-les per temps.

Per tal de trobar conjunts de malalties que són freqüents en un pacient s'aplicaran algorismes de mineria de patrons, donat si s'intentés solucionar d'una manera trivial, el cost computacional seria prohibitiu. Les seqüencies de conjunts impliquen una altre variant en la cerca, una finestra de temps. Aleshores el que resulta és una llista ordenada per instàncies de temps on cada element és un conjunt de malalties que es van diagnosticar a la vegada.

5.1 Conjunts

Trobar conjunts d'elements comuns en un gran nombre d'objectes té moltes utilitats. Un bon exemple, sigui una llibreria amb els registres de venda de llibres a clients, els patrons descoberts són el conjunt de llibres més comprat freqüentment. Amb aquesta informació la botiga pot fer-ho servir per a promocions, posicionament a les estanteries o d'altres.

Donada una matriu M de valors booleans, amb n files on cada fila representa un individu i les columnes són atributs pertanyents a un conjunt A, es pot associar cada individu amb un subconjunt d'A tal que els atributs que conté són certs a la seva fila.

El problema de trobar conjunts freqüents es definiria donat un llindar ε, es volen trobar tots els subconjunts S d'A tals que almenys $\varepsilon \cdot n$ dels individus contenen S. Aquests subconjunts són els que s'anomenen freqüents (depenent implícitament d'ε). El llindar ε és el que es coneix com a mínim suport, explicat a 5.1.1.

Cal veure que el nombre de possibles subconjunts freqüents és $2^{|A|}$. A la pràctica són considerablement menys, però el problema de cercar-los sense fer una cerca exhaustiva no és trivial.

Els algorismes explicats a continuació són els més utilitzats en el camp i cadascun fa servir una aproximació diferent per a trobar els conjunts.

5.1.1 Suport

Els algorismes tant de conjunts com de seqüencies de conjunts, fan servir una mesura per a determinar fins a on agafar els resultats. Aquesta mesura es fa dir mínim suport, indica el percentatge mínim d'aparició respecte les dades donades. El suport d'un conjunt

d'elements és el nombre de vegades que apareix a la base de dades aquell conjunt. En percentatge, es fa sobre el total de les transaccions que hi ha. Sigui per exemple un conjunt amb 500 aparicions entre les dades, si en total hi ha 10.000 transaccions, el suport és del 5%.

$$\mathsf{supp}(A) = \frac{\#\ \text{d'ocurrències d'}A}{\#\ \mathsf{transaccions}} \tag{5.1}$$

5.1.2 Apriori

Aquest algorisme, [AS94], és un dels primers que es van crear per a resoldre el problema de minar regles d'associació sobre grans quantitats de dades on cada un consisteix en un element i la informació de la transacció com ara la data.

La primera passada d'aquest algorisme simplement compta les ocurrències dels elements per a determinar els conjunts freqüents de mida 1. La següent passada consta de dues fases. Primer, els conjunts obtinguts a la passada anterior serveixen per generar els candidats d'aquesta. Sempre a cada passada s'apujara la mida en una unitat. Després, es recorre la base de dades per contar quants dels candidats generats hi són presents. Finalment es treuen els que no compleixen el mínim suport. Es seguiran fent passades fins que no existeixi un conjunt de candidats, és a dir, no es puguin generar més conjunts.

5.1.3 FP Growth

FP Growth, [HPYM04], fa servir una aproximació diferent que l'algorisme Apriori per aconseguir el mateix resultat. Fa servir una estructura anomenada *frequent-pattern tree*, o *FP-tree* abreviat. Aquesta es construeix fent un arbre de prefixos estès on es guarda l'informació rellevant sobre els patrons freqüents, en aquest cas l'identificador de l'element i un enter que representa el compte. Per assegurar que aquesta estructura es compacta i informativa solament hi haurà conjunts freqüents de mida 1 en els nodes de l'arbre. Un cop construït aquest arbre, ja no cal treballar amb tota la base de dades per les següents cerques de patrons freqüents.

L'algorisme comença buscant els patrons amb conjunts de mida 1 i construint aquest *FP-tree*. Aplicant recursivitat es fa una cerca en profunditat per a minar els patrons a l'arbre generat.

5.2 Seqüències

Trobar un conjunt d'elements que en un instant de temps concret són comuns en una gran quantitat d'objectes d'una base de dades té moltes utilitats. Com per exemple, una base de dades d'una web popular enregistra quines pàgines accedeix l'usuari d'aquella web. Els patrons descoberts són seqüències de les pàgines més visitades d'aquella web. Aquesta informació es pot fer servir per a reestructurar la web, o bé per afegir enllaços a altres pàgines segons els patrons d'accés de l'usuari.

Una seqüència és una llista ordenada de conjunts d'elements. Una seqüència s ve donada per $\langle s_1, s_2, \ldots, s_l \rangle$ on s_j es un conjunt. Un element dins d'un conjunt només pot aparèixer una vegada però múltiples al llarg de la seqüència.

Els algorismes que troben aquestes seqüències freqüents es basen en els algorismes per a trobar conjunts freqüents. Com ara el GSP ([SA96]) que es basa en l'Apriori, PrefixSpan ([PHMA+04]) en el FP Growth i l'algorisme SPADE ([Zak01]) amb l'Eclat. El que afegeixen són constriccions de temps.

5.3 Regles d'associació

Un exemple d'una regla d'associació pot ser que el 98% dels clients que compra pneumàtics i accessoris pel cotxe també aprofita per a que li facin algun servei al seu automòbil. Altres aplicacions inclouen disseny de catàlegs, venda d'anuncis, distribució de tendes i segmentació de clients segons els seus patrons de compra.

Una regla d'associació és una implicació de la forma $X \Rightarrow Y$, on X i Y són dos conjunt d'elements que apareix a la base de dades i no hi ha un element en comú entre ells. En aquest cas, X seria l'antecedent de la regla i Y el conseqüent. El suport de la regla és el del conjunt $X \cup Y$, dos altres mètriques importants d'una regla són la confiança i el lift, definits a 5.3.1 i 5.3.2 respectivament.

Donat un conjunt de transaccions $\mathcal{D}$, el problema de minar regles d'associació es generar totes les regles d'associació tals que tenen suport i confiança major que el minim suport (anomenat *minsup*) i mínima confiança (anomenada *minconf*) que l'usuari a especificat respectivament. $\mathcal{D}$ pot esser qualsevol fitxer, base de dades o resultat d'una expressió relacional.

5.3.1 Confiança

La confiança d'una regla d'associació indica la probabilitat de que sigui certa. Sigui una regla $A \Rightarrow B$, si aquests dos elements apareixen en un conjunt 50 vegades i l'element A apareix 100 vegades en tota la base de dades, aleshores la confiança d'aquesta regla és $50/100 = 0.5$, o sigui, el 50% de les vegades que hi ha A també hi ha B en el conjunt.

$$\mathsf{conf}(A \Rightarrow B) = \frac{\mathsf{supp}(A \cup B)}{\mathsf{supp}(A)} \tag{5.2}$$

5.3.2 Lift

El *lift* es una mesura que indica el ràtio de relació entre l'antecedent i el conseqüent d'una regla d'associació, respecte una elecció aleatòria de l'antecedent.

Si el *lift* és 1, probablement l'antecedent i el conseqüent de la regla d'associació són independents entre ells. Això vol dir que la probabilitat de tenir com a antecedent un element o un altre és la mateixa i aquesta regla no serveix. En canvi, si el *lift* és major que 1, l'antecedent i el conseqüent són dependents i per tant l'antecedent si que juga un paper important a l'hora de predir el conseqüent.

$$\mathsf{lift}(A \Rightarrow B) = \frac{\mathsf{conf}(A \Rightarrow B)}{\mathsf{supp}(B)} = \frac{\mathsf{supp}(A \cup B)}{\mathsf{supp}(A) \cdot \mathsf{supp}(B)} \tag{5.3}$$

5.4 Aprenentatge automàtic

Els algorismes d'aprenentatge automàtic generen models capaços de predir nous resultats a partir d'un entrenament previ. És a dir, donades unes dades es vol aconseguir que proporcionant una entrada, el model fa una predicció de la sortida. La fase prèvia d'entrenament serveix per ajustar el model, ja que a l'entrenament se li passa al model les dades d'entrada i la sortida. En el cas d'aquesta aplicació, les dades que es passen d'entrada són l'edat, el gènere i les malalties. El que es vol predir és el nombre de visites que un pacient farà al llarg d'un període específic.

Les prediccions d'un model poden classificar o dónar valors concrets. En aquest cas es podria classificar segons si el pacient ha fet menys de X visites, entre X i Y visites o més de Y. Si es vol fer regressió el model intentarà ajustar els resultats al nombre de visites.

Per aquest projecte s'han utilitzat models d'aprenentatge automàtic per a predir les visites que tindrà un pacient. Els models que s'han triat són: regressió lineal, Lasso, arbres de decisió, SVM, *random forests* i *gradient boosting*. Les explicacions dels algorismes són breus perquè no entra a l'abast del projecte modificar-ne cap, sinó que s'usen com a caixa negra canviant els seus paràmetres únicament. Un bon llibre que els explica en detall és [Bis06].

5.4.1 Regressió lineal

La regresió lineal és un dels models d'aprenentatge automàtic més simples que es fan servir. Assigna a una equació lineal les variables d'entrada i a cada un se li assigna un coeficient w_i, més un biaix (coeficient lliure w_0). El que es vol obtenir (y) és el que es vol obtenir:

$$y(x, w) = w_0 + w_1 x_1 + \ldots + w_D x_D \tag{5.4}$$

Amb aquesta equació s'intenta minimitzar la suma dels quadrats entre les sortides de les dades i els resultats que dóna l'aproximació linear. Cal dir que aquesta estimació que fa

el model és bona quan els termes són independents entre ells. Si no ho fossin s'observaria una gran variància en els resultats.

5.4.2 Lasso

Least absolute shrinkage and selection operator o Lasso és un mètode de regressió lineal que estima coeficients dispersos. A l'hora de crear el model, a l'entrenament, es selecciona un subconjunt de les variables d'entrada per a fer servir en el model final en comptes de fer servir-les totes. Resulta útil per a casos on hi ha una gran quantitat de variables d'entrada i l'únic que fan és afegir soroll al model, fent servir Lasso aquest soroll desapareix ja que s'agafa el subconjunt rellevant per a la predicció.

5.4.3 Arbres de decisió

Els arbres de decisió fan servir l'estructura d'arbre per representar camins de decisió i un resultat per a cada un. Són fàcils d'entendre, interpretar i el procés que segueixen per a arribar a la solució és totalment transparent. Tant poden fer regressió com classificació.

El problema dels arbres de decisió és que trobar un arbre òptim pel conjunt de dades de que es disposa és un problema computacional molt complicat. És molt fàcil sobre-ajustar l'arbre pel conjunt de dades d'entrenament i que després amb les dades que rep més endavant no generalitza adequadament.

5.4.4 SVM

La SVM o *Support Vector Machine* construeix un hiperplà o un conjunt d'hiperplans en espai n-dimensional o infinit el qual és fa servir per a classificació i regressió.

Per a classificar el que vol aconseguir aquest hiperplà és separar dos conjunts de dades diferents amb la màxima distància possible entre els dos. A vegades pot passar que no es separable en aquell espai dimensional, aleshores es crea un espai dimensional amb més dimensions capaç de fer la separació.

A l'hora de fer regressió l'hiperplà que es genera intenta minimitzar la distància entre ell i tots els punts. Per a fer regressió no lineal s'ajuda de paràmetres addicionals que ajuden a modelar l'hiperplà

5.4.5 Random forests

Els *random forests*, en català *boscos aleatoris*, són models de predicció i classificació basats en la construcció de múltiples arbres de decisió. Donat que els arbres de decisió tendeixen a sobre-ajustar, el que busca un *random forest* corregeixi aquest problema.

Donada una base de dades D, es generen arbres de decisió per a entrenar amb diferents versions de la base de dades D modificada de manera que s'agafen uniformement elements d'aquesta podent contenir repeticions. Aquests arbres contenen una mica d'aleatorietat

a l'hora de seleccionar els atributs per on partiran les seves decisions, així doncs, tindran diferents camins de decisió.

Finalment s'agafa una mitja entre totes les prediccions dels arbres entrenats, cosa que ajuda a reduir la variància del model i fer millors prediccions.

5.4.6 Gradient boosting

Es fa servir tant per problemes de classificació com de regressió. El model que genera es basa en un conjunt de models que tenen una predicció inferior però suficient per no ser considerat aleatòria. El que fa és entrenar un model fent servir m mostres de dades amb una distribució de pesos concreta. Seguit, els pesos de les dades que es prediuen pitjor s'augmenten i les que es prediuen millor s'abaixen, aleshores s'entrena un nou model.

D'aquesta manera sempre s'entrenen models fent servir dades que són difícils de predir a les iteracions anteriors, fins que s'obté el model final que engloba tots aquests models entrenats.

5.4.7 Validació creuada

Com a mètode de validació per a entrenar els models, s'ha fet servir validació creuada, en anglès, *cross validation*. Les dades originals es divideixen en particions, segons el tipus de validació creuada que es faci hi haurà més o menys. Un cop es tenen les particions, es selecciona una que servirá per a la predicció i la resta per l'entrenament. Es torna a repetir aquest procés, seleccionant cada cop una partició que no s'hagi fer servir abans per la predicció, fins que totes s'han fet servir per predir una vegada.

El que permet aquest mètode de validació que fa servir totes les dades per a entrenar i predir. Si es disposa d'una base de dades amb poques transaccions cal aprofitar al màxim aquestes dades. Si es separa una gran part per l'entrenament com es faria tradicionalment es perden moltes dades per predir, cosa que si es disposa d'un gran conjunt de dades no és cap problema però si aquestes són limitades fa perdre precisió al model.

6. Funcionalitats de l'aplicació

Aquesta aplicació consta d'unes funcionalitats bàsiques que permenten a l'usuari que les fa servir un anàlisi en profunditat de les malalties que vol analitzar. A continuació s'explica cada una en detall.

6.1 Afegir pacients

La funcionalitat principal que depenen totes les altres, sense pacients i malalties no es pot treure partit de les funcionalitats ja que és la font d'informació. Un cop oberta l'aplicació s'ofereix una opció d'afegir pacients. Cal especificar la ubicació d'un fitxer separat per files on cada una és un diagnòstic d'una malaltia, també cal especificar a quina col·lecció es guardaràn les dades i una etiqueta per a posar a les visites d'aquell dataset. Les dos últimes són opcionals i si no es volen especificar s'agafaràn les que hi ha per defecte.

6.2 Filtrat de pacients per característiques

Els pacients guardats a una base de dades poden ser filtrats segons els seus atributs i més endevant aplicar les funcionalitats que es desitgin. Donat que si es disposa de grans quantitats de dades o bé es vol centrar en un tipus de malatia, interessa que hi hagi una manera de filtrar les dades prèviament. Es proporcionen els següents mètodes per a filtrar i atributs sobre els que es pot filtrar:

Filtres
Igual/Diferent
Major/Menor
Expressió regular
Dins d'un llistat

Atributs
Edat
Gènere
Malaltia
Data de malaltia
Visites

6.3 Visualitzar estadístiques sobre els pacients

L'usuari haurà d'especificar quin atribut vol analitzar dels possibles, com en 6.2, i caldrà especificar quin tipus d'estadística: histograma, *boxplot*, taula, diagrama de barres.

6.4 Generar conjunts i seqüències freqüents

Després de seleccionar quina base de dades fer servir i quins filtres aplicar-li, un cop escollit si es vol generar conjunts de malalties més freqüents o bé seqüències, l'usuari haurà d'especificar quin algorisme vol fer servir (Apriori, FP Growth o Eclat per a conjunts i PrefixSpan, GSP o SPADE per a seqüències) i el mínim suport que en vol emprar. Els algorismes resolen el mateix problema ja sigui de conjunts o seqüències, el que varia és el temps i la memòria usada. Es mostraràn els resultats en forma de taula ordenats per nombre d'elements del conjunt o llargada de la seqüència i en cas d'empat per major mínim suport.

6.5 Generar regles d'associació

Similar a generar els conjunts i seqüències freqüents, aquí l'usuari haurà d'escollir un algorisme, el mínim suport i addicionalment una confiança mínima. Els resultats es mostraràn en forma de taula també mostrant la part esquerra i dreta de la regla, el seu suport, confiança, el suport de la part esquerra i dreta per separat i altres indicadors com el Lift.

6.6 Avaluar models d'aprenentatge automàtic

Aquests models intentaràn predir el nombre de visites de cada pacient, segons un any o bé en total, i mostrarà els resultats en forma de taula amb la predicció i el valor obtingut de cada pacient. L'usuari haurà de triar quin model vol fer servir dels possibles: regressió lineal, Lasso, arbre de decisió, SVM, *random forests*, *gradient boosting*.

6.7 Visualitzar graf de malalties

Aquesta visualització mostrarà la relació que hi ha entre les malalties amb ajuda visual per a poder veure quines són més rellevants. Cada malaltia representa un node en el graf, i una aresta entre dues malalties significa que un pacient ha estat diagnosticat de les dues. El nombre de pacients que tinguin la malaltia influirà sobre el tamany del node així com el tamany de l'aresta segons els parells de pacients que tenen les dues malalties.

6.8 Exportar/guardar resultats

Quan ja s'ha executat una operació amb les dades i es mostren els resultats obtinguts, aquests podràn ser guardats internament per la mateixa base de dades o bé exportar-los a un format CSV o JSON i guardar-ho on es desitgin

7. Implementació de l'aplicació

L'aplicació consta de dues parts diferenciades: el client i el servidor. S'ha triat aquesta estructura per la facilitat d'afegir i treure funcionalitats a nivell de càlcul i per a aprofitar la gran flexibilitat de tenir un client que va fent peticions i s'encarrega de gestionar els resultats que es proporcionen.

Les tecnologies que s'han triat estan basades en web pero el resultat final que s'obté es una aplicació d'escriptori.

7.1 Tecnologies

La part de client s'ha fet amb Electron. Electron és una llibreria de codi obert desenvolupada per Github per a fer aplicacions d'escriptori per a diverses plataformes amb HTML, CSS i Javascript. Fa servir internament Chromium (Versió de codi obert de Chrome) i Node.js, l'aplicació pot ser compilada per a Mac, Windows i Linux. Al fer servir Javascript, s'ha escollit una llibreria per a fer interfícies d'usuari anomenada React desenvolupada per Facebook. Està basat en components, com ara una llista, un element de la llista, un formulari, per després poder composar-los lliurement: Un formulari amb una llista dins i aquesta conté elements.

El disseny del client fa servir un llibreria de CSS que es diu Bulma, i per mostrar les gràfiques Vis.js, una llibreria de visualització de Javascript. S'han triat aquestes tecnologies pel client per la facilitat d'ús i de fer prototips viables en poc temps. Són llibreries populars fetes servir per desenvolupadors de codi obert, *startups* i companyies com ara Github, Microsoft, Slack i moltes més.

Pel que fa al servidor s'ha triat una llibreria en Python, Flask que es minimalista i fàcil d'usar per a fer prototips ràpids. A més també es fa servir una altra llibreria de Python per a fer l'aprenentatge automàtic i els processos entremitjos com ara llegir les dades entrants, transformació a graf, passar als formats necessaris pels algorismes.

Algorismes

S'ha escollit fer servir llibreries que ja implementin els algorismes descrits en les seccions 5 i 5.4. Això es degut a una cerca prèvia d'algorismes disponibles, els quals es va trobar dues llibreries per a patrons freqüents que són Coron i SPMF que implementaven els algorismes que es volien fer servir de manera fiable i eficient en Java. De mateixa manera, per a fer aprenentatge automàtic hi ha una llibreria de Python que destaca: Scikit-learn, disposa de una bona documentació, exemples i té una gran comunitat de programadors al darrere.

Per a la persistència de dades, es fa servir MongoDB, una base de dades NoSQL orientada a documents. Donat que es tracta d'un primer prototip d'aplicació, la flexibilitat de MongoDB fa que es pugi canviar fàcilment el model de dades o fins i tot no tocar-ho i entrar les dades en un document de manera diversa.

7.2 Client

El client es el que s'encarrega de llegir les dades d'entrada i passar-les al servidor que les processara. S'han dissenyat diferents vistes per a poder fer l'anàlisi de les dades. A continuació es descriu cada fase que te el client.

7.2.1 Entrada de dades de pacients

El primer de tot que apareix a l'aplicació, es la selecció de base de dades. Si ja existeix una, especificar quina i sinó cal introduir un fitxer Excel com es descriu a la secció 8 i a quina col·lecció es guardarà. Altrament, si ja s'ha afegit un o varis fitxers Excel aleshores es pot seleccionar aquella base de dades per un nom assignat.

Si s'escolleix guardar, hi ha un camp on s'ha de seleccionar el fitxer, un altre pel nom de la col·lecció que es vol fer servir (pot existir prèviament) i un camp per etiquetar les visites. El fitxer s'envia al servidor, que és l'encarregat de recòrrel i guardar finalment a una col·lecció de MongoDB especificada pel nom donat i guarda les visites al pacient amb l'etiqueta que s'ha escollit.

Si es fa servir una base de dades existent, es selecciona la col·lecció que es desitja. D'una manera o altra, es guarda internament aquest nom de la base de dades per a les posteriors consultes.

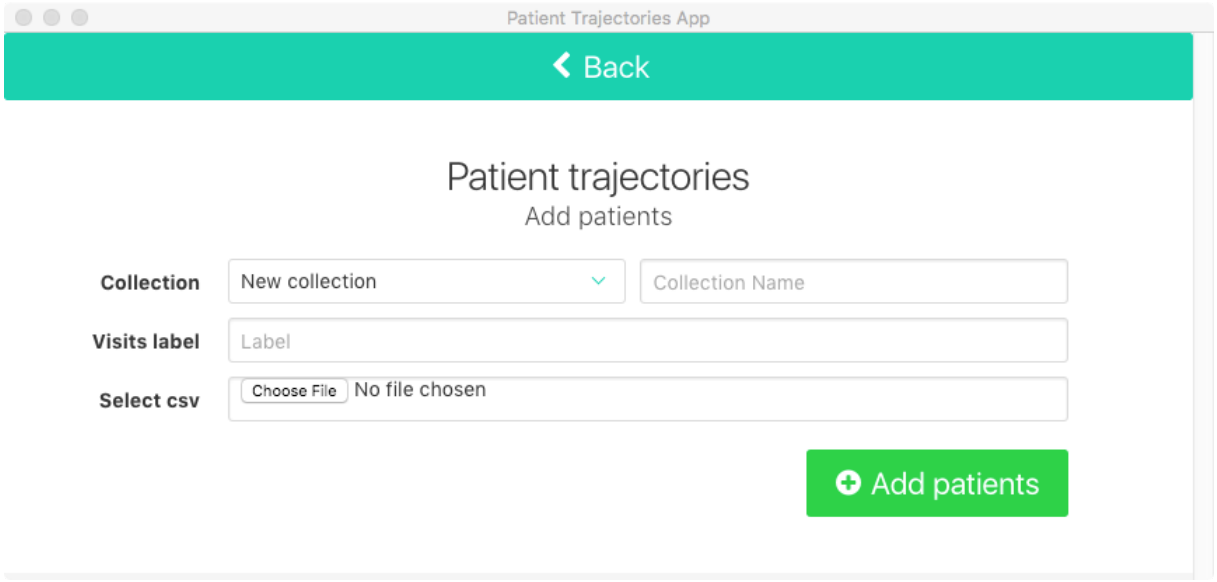


Figura 7.1: Entrada de dades.

7.2.2 Filtres

Un cop seleccionada la base de dades amb què treballar, es poden afegir filtres com ara gènere, edat o excloure malalties. L'interfície permet anar afegint filtres, primer no hi ha cap disponible, s'ha de prémer el botó d'afegir filtre aleshores apareix una fila amb les entrades següents:

- **Camp**: Seleccionable amb els possibles atributs que es poden fer servir per filtrar.
- **Filtre**: Seleccionable amb els possibles tipus de filtres.
- **Valor**: Camp de text on s'entra el valor pel que es vol filtrar.

Es poden anar afegint lliurement, el que no es controla que hi hagi contradiccions entre els filtres, en cas d'haver-n'hi, simplement els resultats sortiràn buits. Aquests filtres es guarden internament a l'aplicació per després utilitzar-los o modificar-los.

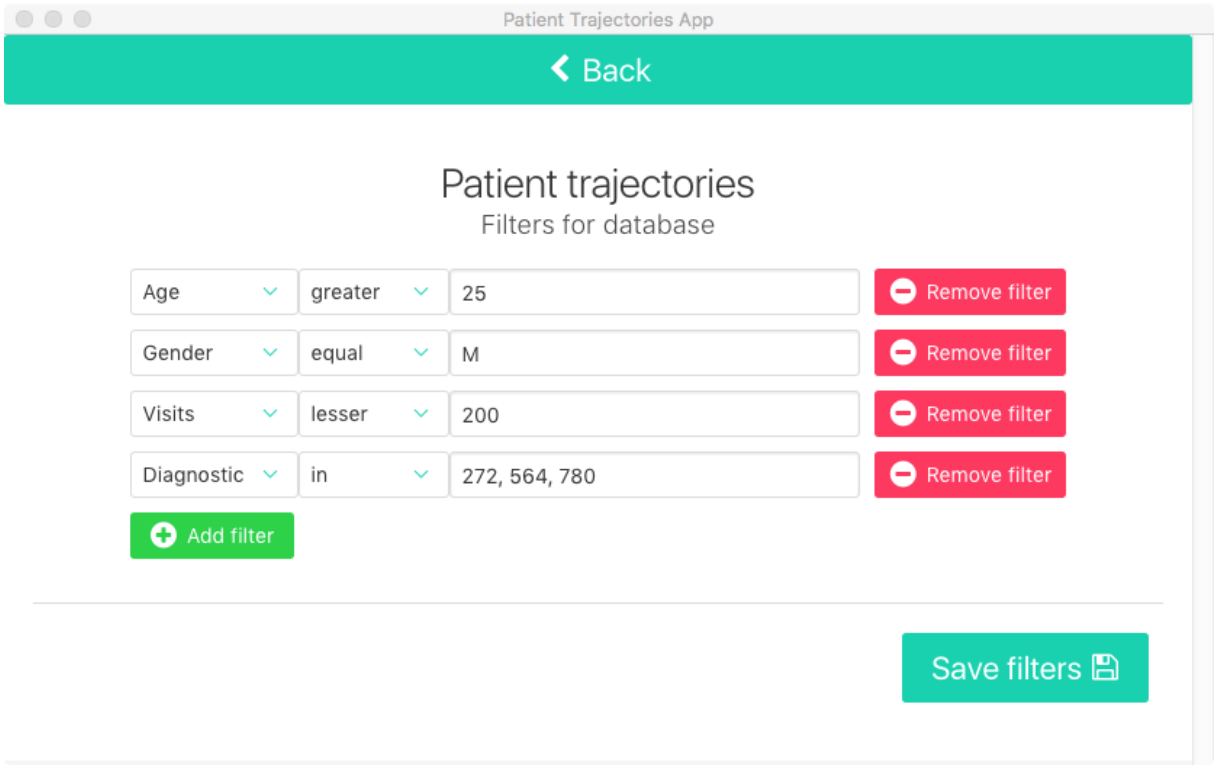


Figura 7.2: Filtres sobre la base de dades.

7.2.3 Càlculs

Es pot triar entre diferents càlculs que fer a les dades. Un cop llegides i filtrades les dades, l'usuari pot triar entre les següents opcions:

- **Estadístiques**: Cal especificar el tipus de gràfic a mostrar: Boxplot, Histograma o Diagrama de barres. Després hi ha una selecció on es pot triar l'atribut.
- **Conjunts freqüents**: Selecció de l'algorisme i el mínim suport que es vol observar.

- **Seqüències de conjunts freqüents**: Selecció de l'algorisme i el minim suport que es vol observar.
- **Regles d'associació**: Selecció de l'algorisme, el mínim suport i la confiança per a la regla.
- **Aprenentatge automàtic**: Selecció de l'algorisme.
- **Graf de relacions entre malalties**: Entrada númerica amb el mínim de pacients que tenen les dues malalties (arc).

Un cop entrades les dades pel formulari aquestes s'envien al servidor amb la base de dades seleccionada al principi i els filtres escollits, passant a la següent vista on es mostren els resultats obtinguts per aquella aplicació.

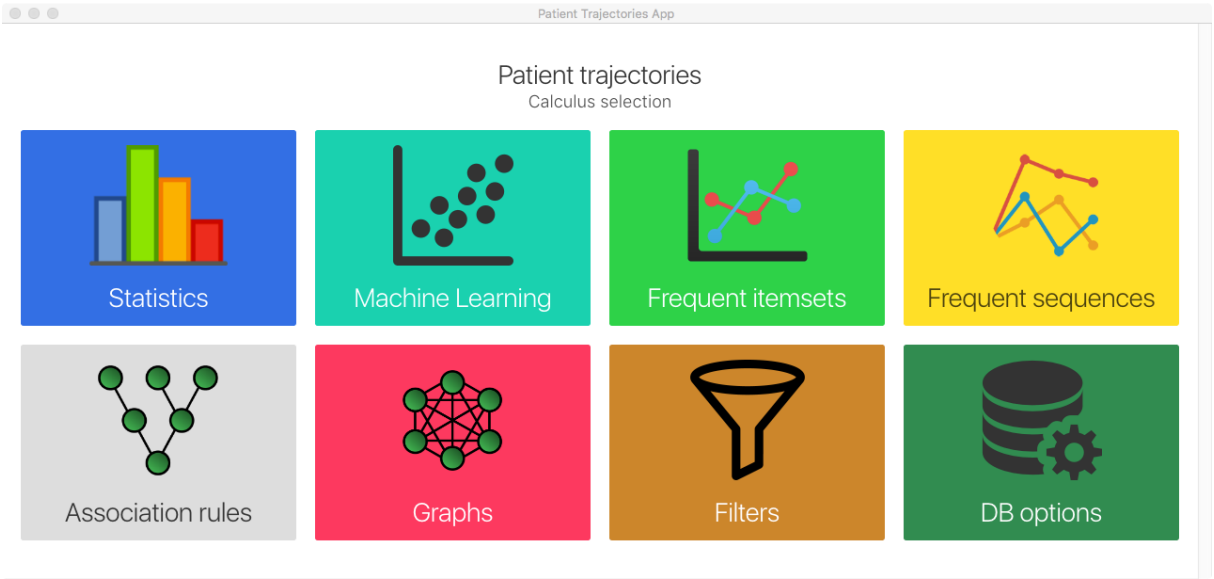


Figura 7.3: Vista on es mostren els càlculs.

7.2.4 Resultats

Els resultats obtinguts de les aplicacions es mostraràn de manera gràfica i/o en taula. Les aplicacions de conjunts, seqüencies freqüents i regles d'associació mostren una taula, les estadístiques, aprenentatge automàtic i graf mostren taula i gràfic.

Aquests resultats podran ser guardats a la base de dades, exportats a un document separat per comes (*CSV*) o be en *PNG* si és una imatge.

7.3 Servidor

El servidor es el que s'encarrega de rebre les peticions del client i fer els càlculs pertinents per a generar els resultats. Està organitzat per *endpoints* on a cada un se li pot demanar una tasca diferent. Les peticions es poden realitzar mitjançant el protocol d'internet *Hypertext Transfer Protocol* (HTTP/1.1) amb el mètode `POST` pel pas de dades.

Els *endpoints* comparteixen diversos requeriments. A les peticions sempre hi haurà la part del cos del missatge un camp amb `collection` i un altre amb `query` que representa la base de dades d'on agafar els pacients i el filtre que aplicar per treure els resultats. Després, segons la petició que s'hagi fet es comprovaràn la resta de paràmetres. Finalment els resultats es retornen al client en forma de `JSON`, un format fàcil de manipular amb Javascript.

Cal dir que les llibreries triades per a mineria de patrons freqüents utilitzen un format de dades específic, el servidor és el que s'encarrega de transformar les dades dels pacients al format adequat i aleshores cridar a les llibreries SPMF i Coron. A continuació es fa una breu descripció d'aquests formats.

7.3.1 Estructura de dades rebudes

Les dades que es llegiran per a guardar en el servidor hauran de tenir un format Excel amb un diagnòstic per fila amb les següents columnes:

- **CODIPACIEN**: Identificador de pacient.
- **EDAD**: Edat.
- **SEXO** Gènere.
- **DATAINICIO**: Data d'inici quan es va diagnosticar la malaltia.
- **DATAFIN**: Data final quan es va curar. Pot ser buida.
- **CODICORRECTECIM**: Codi CIM de la malaltia que s'utilitzarà.
- **DIAGNOSTIC**: Descripció de la malaltia.
- **NUMVISITAS**: Número de visites relatives a aquell Excel.

7.3.2 Formats d'entrada de les llibreries

Fent servir la taula de la figura 7.1 com a exemple, s'explicaran els formats que fan servir les llibreries utilitzades Coron i SPMF. Donats els objectes O_i, cadascun té uns atributs marcats amb una X, si té l'atribut $a(1)$, aquest tindrà una creu en la seva columna. A partir d'aquí, les dades poden agafar tres formats diferents anomenats Basenum, Bool i RCF.

Basenum

Consisteix en document de files on cada una conté una seqüència d'enters tals que representen una característica que té la fila. A una mateixa característica se li assigna un enter de manera que si diferents files tenen el mateix atribut aleshores apareixerà el mateix enter a la fila.

En el cas de l'aplicació cada fila representa un pacient i una característica pot ser una malaltia, edat, sexe o visites.

	a(1)	**b(2)**	**c(3)**	**d(4)**	**e(5)**
O_1	X	X		X	X
O_2	X		X		
O_3	X	X	X		X
O_4		X	X		X
O_5	X	X	X		X

Taula 7.1: Taula amb les dades originals

Basenum per a seqüències

Cal afegir entre seqüència i seqüència d'enters un -1 i al final de cada fila un -2. Aquest format es fa servir per a mineria de seqüències de conjunts freqüents.

Bool

En aquest cas un document és un fitxer amb zeros i uns (però no binari) on cada fila apareixen totes les característiques en un mateix ordre de manera que si aquella fila conte l'atribut, aleshores hi haurà un 1 en aquella posició, altrament un 0.

Aquest format pot resultar poc eficient si hi ha moltes característiques i el pacient només posseeix unes poques, el que faria un document amb molts zeros i pocs uns (dispers).

RCF

És molt similar al format Bool però té l'avantatge que es poden anomenar els objectes i els atributs. Cada fila representa un objecte, amb l'ordre en que apareixen llistats i a cada columna l'atribut també en l'ordre llistat.

7.3.3 Endpoints

Els *endpoints* representen peticions al servidor, el text en parèntesi indica quina URL fa servir l'aplicació per connectar-s'hi. Totes aquestes crides es fan mitjançant el mètode `POST`.

Guardar dades (`/store`)

S'obre i es llegeix el fitxer amb les dades, que es processen fila a fila, agafant les columnes especificades, i així generant un document per cada pacient amb les seves malalties agrupades. Al número de visites se li afegeix al camp `num_visits` amb l'etiqueta especificada. Finalment s'afegeix a la base de dades seleccionada, tal que si ja existeix el pacient, només afegeix els nous codis de malaltia i les visites. S'assumeix que la resta de dades son correctes i per tant no varien (edat, sexe) i els codis de les malalties no són repetits.

Basenum	**bool**	**rcf**
1 2 4 5	1 1 0 1 1	[Relational Context]
1 3	1 0 1 0 0	Default Name
1 2 3 5	1 1 1 0 1	[Binary Relation]
2 3 5	0 1 1 0 1	Name_of_dataset
1 2 3 5	1 1 1 0 1	$o1 \mid o2 \mid o3 \mid o4 \mid o5$
		$a \mid b \mid c \mid d \mid e$
		1 1 0 1 1
		1 0 1 0 0
		1 1 1 0 1
		0 1 1 0 1
		1 1 1 0 1
		[END Relational Context]

Taula 7.2: Conversions a basenum, bool i rcf de la taula 7.1

Generar conjunts (`/itemsets`)

Es rep una petició amb l'algorisme a utilitzar i el mínim suport que es vol. Es fa la consulta a la base de dades de pacients (amb els filtres que s'hagin especificat) i aleshores es passa al format de dades adequat, per a la llibreria SPMF es fa servir Basenum i per Coron el format RCF. Des del servidor en Python es crida a la llibreria en Java després de comprovar que els paràmetres són correctes. Els resultats parcials es guarden a una carpeta temporal, es llegeixen i es transformen al format de sortida (*JSON*).

Generar seqüències (`/sequences`)

La generació de seqüències és molt similar a la de conjunts, es fa servir únicament la llibreria SPMF i el format és el Basenum per a seqüències. S'agafa de la petició el nom de l'algorisme a fer servir i el mínim suport. Cal filtrar tots aquells pacients que no tenen malalties amb una data.

Generar regles d'associació (`/rules`)

Les regles d'associació es generen utilitzant algorismes que busquen conjunts freqüents de la llibreria Coron. Es fa servir el format RCF per a entrar les dades a l'algorisme. S'agafa el nom de l'algorisme, el mínim suport i la confiança.

Aprenentatge automàtic (`/ml/:model`)

En aquesta petició es passa el model a utilitzar a la URL, pel cos del missatge es passen els paràmetres addicionals específics de cada model. Per a entrar les dades als models, fa falta passar les dades a un vector de vectors on cada vector representa un pacient. Les dades es passen a un format similar a Bool, però en comptes de files hi ha vectors de pacients. S'ha creat un modul on s'encapsulen les funcions d'entrenament i generació de models fent servir la llibreria Scikit-learn. Totes fan servir validació creuada, fent *10-fold cross validation* es fan aleatòriament 10 particions de la mateixa mida dels pacients obtinguts. Aleshores cada partició es fa servir per provar el model una vegada i 9 per fer l'entrenament del model.

Estadístiques (`/statistics/:plot`)

Aquest punt rep el tipus de gràfica que es vol visualitzar a través de la URL i quin atribut de les dades cal agafar per a fer la representació.

Graf (`/graph`)

El graf de les malalties rep el mínim pes que es necessita per a mostrar una aresta. Un cop extrets els pacients cal construir el graf de malalties. Es construeix mapa (diccionari en Python) amb el parell de malalties com a clau i el nombre de pacients que li han diagnosticat les dues com a valor. Es recorre per cada pacient les seves malalties i per cada una les següents de la seva llista, de manera que si s'ordena el parell de malalties es tenen tots els parells de malalties que el pacient pot generar. Si en el mapa ja existeix la parella, se li suma 1 al valor, altrament es crea de nova i s'assigna un 1.

El graf s'envia en format `JSON` al client en forma de llista on cada element és un objecte amb el parell de malalties i el nombre de pacients.

8. Anàlisi d'un conjunt de dades

Les dades que s'han analitzat han estat proporcionades per Serveis mèdics, en format Excel on cada fila hi ha un diagnostic de la malaltia a un pacient amb les columnes descrites a la secció 7.3.2.

Per cada any hi ha un arxiu, des del 2010 fins al 2015 i un arxiu amb tots els diagnòstics anteriors a 2010. En total, uns 32.000 diagnòstics que un cop carregats a l'aplicació pertanyien a uns 9400 pacients. Els diagnòstics amb un codi de malaltia incorrecte es descarten.

Aquests codis estan definits en un format ICD-9-CM (*International Classification of Diseases, Ninth Revision, Clinical Modification*). Aquest és els sistema oficial per assignar codis a diagnòstics i procediments fets servir a un hospital. Aquests codis poden contenir una lletra majúscula opcionalment, seguit de 3 xifres i poden contenir un o dos dígits més separats per un punt si són més específics.

Per fer aquest anàlisi, per consell de la doctora Juliana Ribera, s'ha concentrat en els pacients que tenen una de les 20 malalties considerades que tenen més impacte. La doctora va proporcionar un document amb una classificació de codis de diagnòstics amb la corresponent malaltia, donat que agrupant aquests codis i posant una etiqueta a les malalties ajuda a l'hora de fer l'anàlisi final. Es veu amb més claredat i aporta millors resultats que d'altra manera, amb un conjunt de codis que representen la mateixa malaltia, no hagués estat possible de deduir.

Aquestes 20 malalties es considera que tenen un impacte més gran en el sistema de salut per la seva llarga durada, cronicitat i cost en el tractament. Els diagnòstics que no apareixen al document igualment han estat inclosos, encara que no s'ha fet una agrupació. La tria d'aquestes 20 malalties no és una elecció sense fonaments, és bastant estàndard i apareix en altres estudis, com per exemple a [RBLA15].

A continuació es mostren les classificacions que s'han fet pels codis dels diagnòstics. Els codis que no pertanyen a aquestes malalties s'han generalitzat agafant només la part sense els dígits específics, és a dir, la part sense el punt i els següents dígits, de manera que es fa una agrupació major.

Malaltia	Codis digitals ICD-9-CM
Hipertensió	401.0 - 401.9
Insuficiència renal i transtorns de bufeta	584.0 - 586.9, 403.90, 639.3, 669.3
Afeccions pulmonars (incloent malalties pulmonars cròniques) - asma	416.0 - 416.9, 493.0 - 493.92 , 518.81 - 518.84
Infart de miocardi	410.0 - 410.92
Diabetis	250.00 - 250.03, 250.10 - 250.93
Càncer	140.0 - 239.9
Malaltia vascular perifèrica	443.0 - 443.9
Malaltia cerebrovascular (atac de cor)	436.0 - 438.9
Afeccions neurològiques (incloent Parkinson, epilèpsia, MS)	332.0 - 333.4, 345.0 - 345.91
Glaucoma	365.0 - 365.9
Osteoporosi	733.0 - 733.99
Depressió	296.3 - 311.9
Obesitat	244.9, 259.9, 278.00 - 278.8
Abús de drogues (incloent el tabac)	291.0 - 292.9, 305.00 - 305.93
Demència i Alzheimer	294.0 - 294.9, 331.0, 331.19
Insuficiència cardíaca (incloent congestiva)	398.0 - 398.9, 402.0 - 402.9,425.0 - 425.9, 428.0 - 428.9
Artritis reumatoide	714.00 - 714.9
Teixit connectiu (incloent artritis)	716.00 - 716.99
Paraplegia	344.0 - 344.9
Altres condicions cròniques	472.0 - 472.2, 577.1, 601.1, 730.1

Taula 8.1: Malalties segons els codis ICD-9-CM que tenen més impacte

Per a aconseguir un major suport amb aquestes malalties, s'ha agafat pacients que se'ls hi va diagnosticar amb almenys una malaltia de la taula 8.1. Les altres malalties distorsionarien l'anàlisi, perquè o bé són banals, com ara la grip, lesions musculars o traumatismes que poden passar aleatòriament, encara que el pacient pateixi una malaltia complexa, o bé perquè no són tant corrents. Encara que poden seguir apareixent, el assegurar que existeix una de les 20 malalties triades fa que l'anàlisi tingui més coherència. Això ha fet reduir els pacients per analitzar a 3748 exactament. Encara que s'hagi abaixat el número els resultats seran més clars donat que si es vol veure quines relacions tenen aquestes malalties, els pacients que no tenen cap ni una diagnosticada, només fan que afegir soroll.

Donat que la cerca de patrons freqüents intentat relacionar l'edat i les visites al metge amb les malalties és molt difícil que surti una relació rellevant amb una sola edat o número de visites concret, s'ha estratificat les edats segons la taula 8.2 i el número de visites classificant per dècimes, *0-9* visites, *10-19* visites, etc.

Edats
Menor o igual que 45 anys
Entre 46 i 65 anys
Entre 66 i 74 anys
Igual o major que 75 anys

Taula 8.2: Estratificació de les edats

8.1 Conjunts obtinguts

Fent servir l'algorisme Apriori amb un mínim suport del 2%, és a dir, hi ha almenys 75 pacients que presenten aquest conjunt de malalties respecte el total de 3748. A la taula 8.3 es mostren alguns dels conjunts més rellevants que s'han pogut trobar. L'utilització d'altres algorismes per a trobar conjunts a resultat similar, Eclat, FP Growth no han mostat millors conjunts i suports.

8.2 Seqüències obtingudes

Obtenir seqüències és més complicat perquè requereix d'una data de temps. Amb les dades proporcionades, moltes dates eren buides o errònies, indicant que s'havia diagnosticat l'any 0, segurament posat per defecte pel programa. Fent servir l'algorisme GSP amb mínim suport $0,3\%$, que és on s'han pogut extreure algunes seqüències.

Les seqüències obtingudes són similars als conjunts. A la taula 8.4 es poden veure algunes de les seqüències de conjunts obtingudes. Els resultats amb altres algorismes que cerquen seqüències freqüents han obtingut resultats similars, SPADE, PrefixSpan, LAPIN. Cap d'ells ha estat capaç de trobar millors seqüències.

Conjunt	Suport
Hipertensió, Transtorns en el metabolisme, Diabetis	3%
Hipertensió, Transtorns en el metabolisme, Depressió	2,9%
Hipertensió, Transtorns en el metabolisme, Diabetis	3%
Hipertensió, Transtorns en el metabolisme, Obesitat	2%
Hipertensió, Depressió	6,8%
Depressió, Febres, marejos i col·lapses	6,1%
Diabetis, Transtorns en el metabolisme	5%
Obesitat, Transtorns en el metabolisme	3,8%
Hipertensió, Artrosis	3%
Càncer, Febres, marejos i col·lapses	2,1%

Taula 8.3: Conjunts freqüents de malalties

Seqüència	Suport
Transtorns en el metabolisme → Diabetis → Hipertensió	0,35%
Càncer → Transtorns en el metabolisme → Depressió	0,35%
Trastorns funcionals intestinals i digestius → Depressió	1,52%
Diabetis → Transtorns en el metabolisme	1,5%
Depressió → Refredat comú, nasofaringitis	1,3%
Obesitat → Febres, marejos i col·lapses	0,7%

Taula 8.4: Seqüències freqüents de malalties

8.3 Regles d'associació

Les regles d'associació han resultat coherents. Fent servir l'algorisme d'Apriori, buscant regles amb un mínim suport del 2% i una confiança mínima del 20%. La confiança es calcula dividint el suport de la regla entre el suport de la part esquerra de la regla. De manera que totes les transaccions que contenen la part esquerra de la regla tenen una probabilitat igual a la confiança de contenir la part esquerra.

Regla	Confiança	Suport
Pròstata ⇒ Home	94,52%	5,52%
Taquicàrdies i arritmies ⇒ Hipertensió	67,67%	2,4%
Trastorns funcionals intestinals i digestius ⇒ Depressió	61,9%	3,82%
Transtorns en el metabolisme, Diabetis ⇒ Hipertensió	59,79%	3,01%
Hipertensió, Diabetis ⇒ Transtorns en el metabolisme	57,95%	3,01%
Diabetis ⇒ Hipertensió	53,13%	5,2%
Hipertensió, Obesitat ⇒ Transtorns en el metabolisme	52,38%	2,05%
Artrosis ⇒ Hipertensió	51,83%	3,01%

8.3.1 Discussió dels resultats

Els resultats obtinguts amb aquests algorismes en el conjunt de dades proporcionades ha estat una mica decebedor. És normal que les regles i seqüències més freqüents corresponguin a relacions ben conegudes (per exemple, entre els símptomes de l'anomenat "síndrome metabòlica": diabetis, obesitat, dislipèmia, hipertensió, etc.) . Però s'esperava alguna regla menys corrent i més interessant ocasionalment, així com més regles i patrons en general.

Una explicació plausible és el fet que Serveis Mèdics només proporciona atenció primària, no hospitalària, i per tant els diagnòstics i tractaments a l'hospital no estan enregistrats. A més, Serveis Mèdics treballa per a asseguradores a les quals estan inscrits els seus pacients, que poden triar entre diferents serveis per a ser atesos quan tenen algun problema. Per tant, les dades disponibles a Serveis Mèdics són només una part dels problemes de salut dels pacients atesos.

Finalment, el volum de dades disponible és moderat.Hi ha estudis en que tenint dades d'atenció primària i d'hospital conjunta han aconseguit extreure bons resultats, com en els casos de [CSPI+14], [CSPI+16] i [CNRM14], amb els dos primers fent servir dades del sistema català, però fent servir una magnitud molt superior de dades que les analitzades en aquest projecte.

8.4 Models d'aprenentatge automàtic

L'aplicació de models d'aprenentatge automàtic per a predir el número de visites que un pacient arribarà a tenir no ha estat satisfactori. Els models obtinguts presentaven una gran variància, gairebé igual a la mitjana de visites. Provant tots els models possibles, cap d'ells ha estat capaç de predir significativament millor que els altres. El que indica

que les dades no contenen prou informació per a obtenir models útils.

Per a veure que l'aplicació del projecte funcionava correctament i els algorismes utilitzats eren fiables, es va fer un estudi més exhaustiu per tal de veure si es podia extreure models de predicció efectius d'aquest conjunt de dades proporcionades o bé realment no es pot aconseguir cap model satisfactori.

Se sospita que el motiu pot ser el que s'ha explicat anteriorment: les dades reflecteixen només una part dels problemes de salut dels pacients, només una part de les visites d'atenció primària i cap de les visites hospitalàries que es puguin haver fet.

8.4.1 Cost vs Complexitat

Donades les dades amb els pacients que presenten almenys alguna de les malalties de la taula 8.1, es vol veure si existeix alguna relació entre el cost que representa un pacient i la complexitat de malalties que pateix.

En aquest cas el cost d'un pacient representarà les visites que ha tingut, seria millor una dada econòmica però no es disposa de res millor en aquestes dades. La complexitat del pacient vindrà donada pel nombre de malalties, sumant 1 per cada malaltia. Donat que puntuar o assignar pes a una malaltia per la seva gravetat és un problema en si per a un treball, s'ha optat per fer-ho simple, en el cal que es pogués intuir un camí a seguir es faria més complicat.

Per veure si hi ha diferències entre malalties comuns i les de gran complexitat, s'ha agafat 3 malalties complexes com són la diabetis, el càncer i la hipertensió per a veure si es poden apreciar. S'ha classificat de manera que si un pacient patia de diabetis, se l'afegia en aquell conjunt, si no en tenia i patia d'hipertensió, aleshores es marcava com a aquell grup. Finalment els que no tenien cap de les dues i patien càncer, se'ls posava en aquell grup. Finalment la resta que no tenien cap de les tres escollides es posava en un altre grup. Un cop feta la classificació en 4 grups, es calculava una regressió lineal per a cadascun per veure si havia diferències entre les rectes de regressió.

Com es pot veure a la figura 8.1, les rectes de regressió són molt similars per no dir iguals. Les línies de punts vermelles que creuen els eixos representen la mitjana de cost i complexitat respectivament, tal que s'esperaria que les malalties de més complexitat es situessin en majoria al quadrant superior dret. En aquest cas no sembla que sigui així, hi ha punts de tots colors dispersos per tots els quadrants.

Per acabar de comprovar que no hi havia cap resultat significatiu, es va augmentar el cost de les malalties triades (Diabetis, Hipertensió i Càncer) en 10 punts si el pacient les patia. Però com es pot apreciar a la figura 8.2 hi ha un desplaçament clar de 10 unitats a les malalties triades, alguna de 20 amb diabetis i hipertensió que poden tenir dues malalties complexes i diabetis únicament a la zona de cost 30 amb alguns punts allunyats que no representen grans canvis. Les rectes de regressió ara es poden diferenciar però es degut a aquest desplaçament provocat.

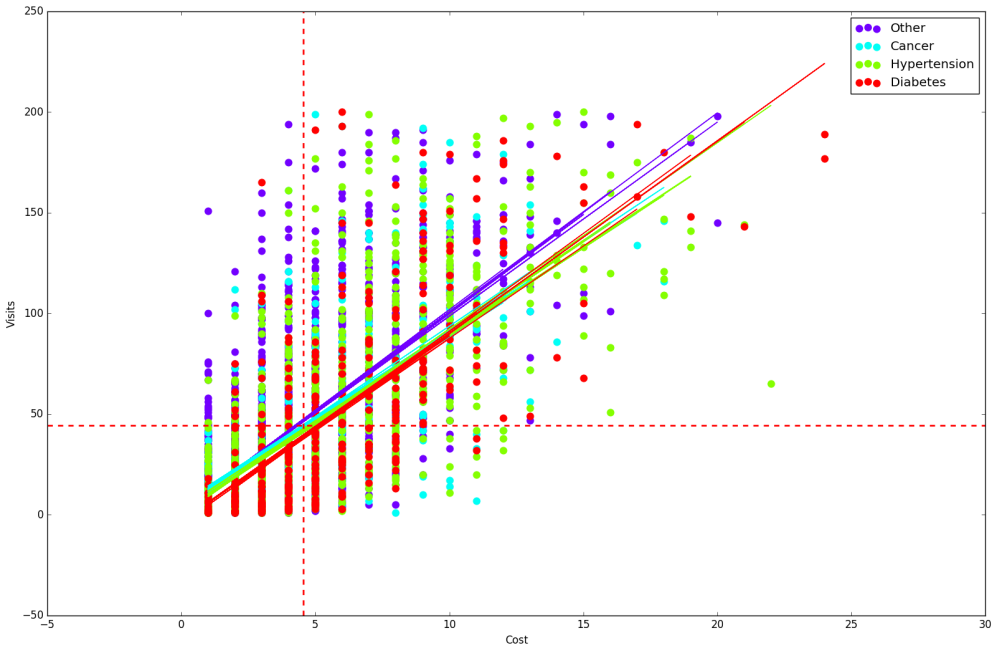


Figura 8.1: Cost vs Complexitat.

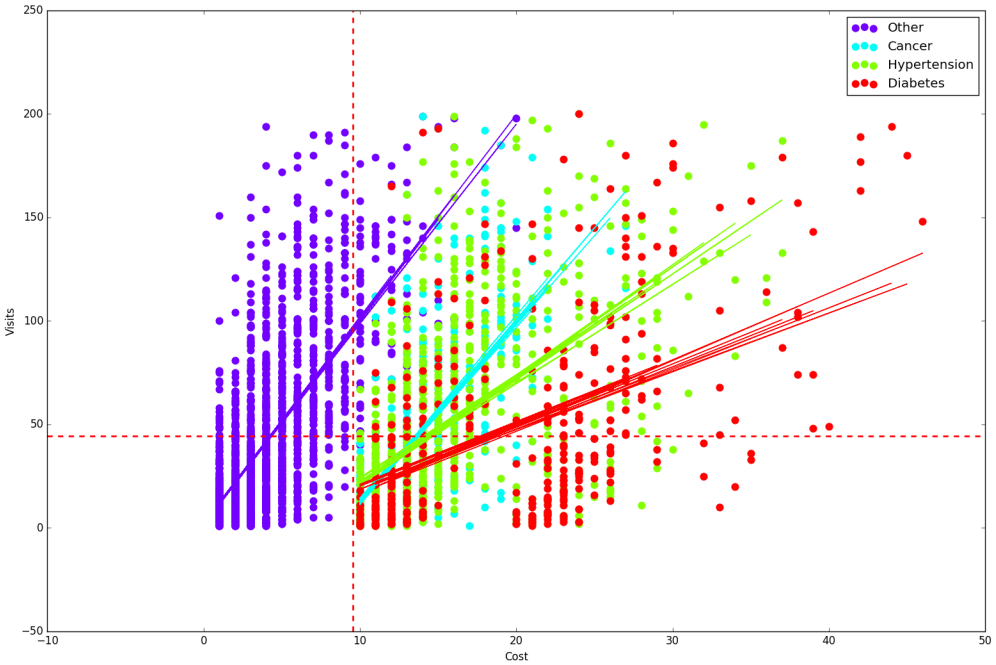


Figura 8.2: Cost modificat vs Complexitat.

8.4.2 Predicció segons l'any i diferents variants

Donat que la predicció de visites totals d'un pacient respecte les malalties, edat i gènere no estava resultant efectiva, es va buscar de predir les visites d'una altra manera. Donat de que es disposava de les visites del 2010 fins al 2015 de pacients, es va intentar de predir les visites de 2015 en funció de les malalties i les visites dels anys anteriors.

Es pot veure a la figura 8.3 les prediccions de visites totals pels models de regressió lineal, Lasso, arbres de decisió, SVM, Random Forests i Gradient Boosting. Tots mostren una línia recta diagonal perquè s'està mostrant la gràfica d'errors, a l'eix de coordenades apareixen els valors mesurats, reals, i a l'eix d'ordenades hi ha les visites predites. Com es pot veure no segueix la recta, que seria l'error mínim quan la mesura real i la predita coincideixen. El problema segueix apareixent a la figura 8.4 amb la predicció de visites per l'any 2015, el que hi ha menys dades perquè hi ha pacients que no tenen enregistrades visites aquell any.

Es pot veure que hi ha algunes que perfilen millor la recta en canvi d'altres que són totalment rectes com en les SVM. Vist que no s'obté cap millora de models, cal veure si hi ha alguna altra manera de potenciar els models.

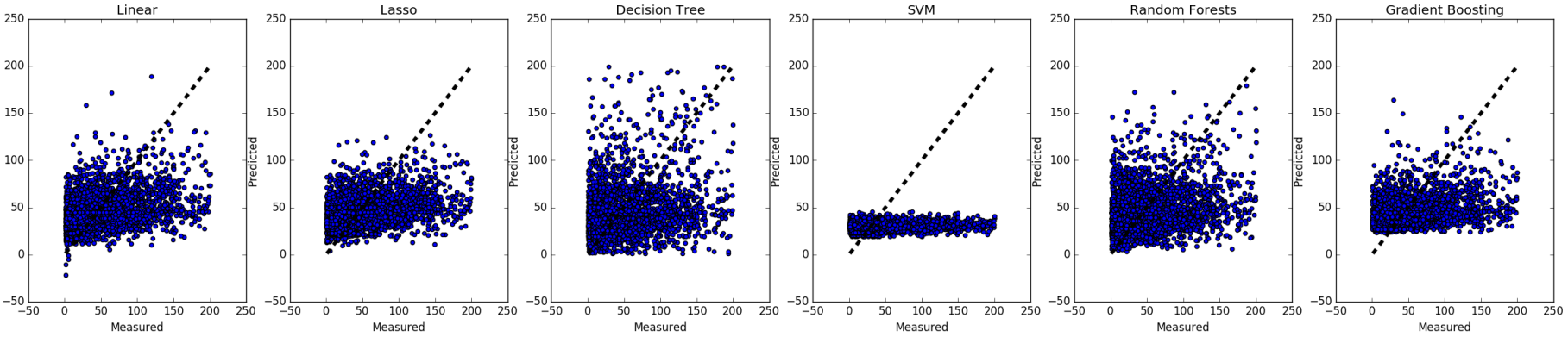


Figura 8.3: Errors de predicció de visites totals.

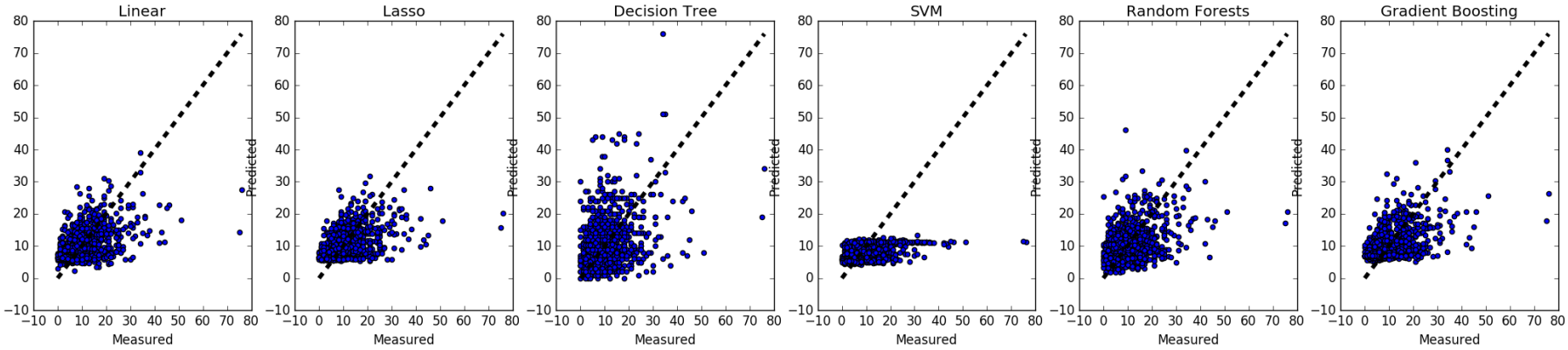


Figura 8.4: Errors de predicció de visites en el 2015.

Variants

- Una primera idea va ser afegir les regles d'associació obtingudes com a indicadors, afegint les mateixes columnes que regles generades. Per a cada pacient es marca quines regles d'associació conté, a cada columna corresponent. Aquesta prova es va fer solament amb el model Lasso, donat que amb els altres models representava un càlcul que no era immediat.
- Una altra idea senzilla, va ser fer servir només les 20 malalties de la taula 8.1 com a variables per veure si era qüestió que es sobresaturava els models. Tenint menys variables, les que resten agafen un paper més important en la predicció.
- La combinació de les dues idees anteriors, agafar només com a variables les 20 malalties sumant les regles d'associació.
- Introduir una nova mesura per a classificar les regles, el *lift*, explicat a la secció 5.3.2. Agafant solament les 20 malalties més aquelles regles d'associació que tenen un *lift* superior a 2. El *lift* major a 1 indica que aquella regla, sigui $A \Rightarrow B$, té més probabilitat de succeir donat A que altres parts esquerra que acabin en B. Agafant aquells que tenen un lift superior a 2 pot ajudar a treure diferències i generar millors models.
- Un altre possibilitat seria agafar solament les regles d'associació obtingudes.
- Finalment fer servir les regles d'associació amb *lift* major que 2 solament.

A continuació a la figura 8.6 es mostra per aquest ordre les variables utilitzades: Totes les malalties, totes les malalties amb les regles d'associació, solament les 20 malalties de la taula 8.1, les 20 amb les regles d'associació, les 20 amb les regles d'associació que tenen un *lift* major a 2, solament totes les regles d'associació i per acabar les regles d'associació que tenen *lift* major a 2. De dalt a baix, a la primera fila es fa servir el model Lasso, a la segona arbres de decisió, a la tercera SVM, a la quarta Random Forests i l'última fila Gradient Boosting.

Com es pot apreciar, no hi ha una recta de punts que s'aproximi a la diagonal. En el cas de Lasso per a totes les malalties sembla que comenci a agafar forma peró l'error que s'obté es superior a les 20 visites, que fa o no fa és la mitja de visites. En el cas de les SVM, totes les prediccions han resultat anar a parar a un espai entre 25 i 50 visites per a qualsevol entrada. Els altres models presenten molta variabilitat i no són usables.

Doncs, cal concloure que cap dels models han aconseguit ser bons, encara que s'intentin diferents aproximacions cap d'aquestes ha resultat i fa pensar que les dades no són suficients per aconseguir resultats significatius.

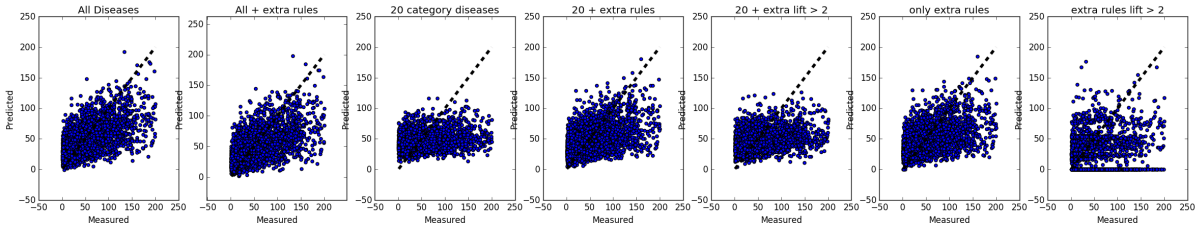


Figura 8.5: Errors de predicció per a tots els models fent servir diferents estratègies.

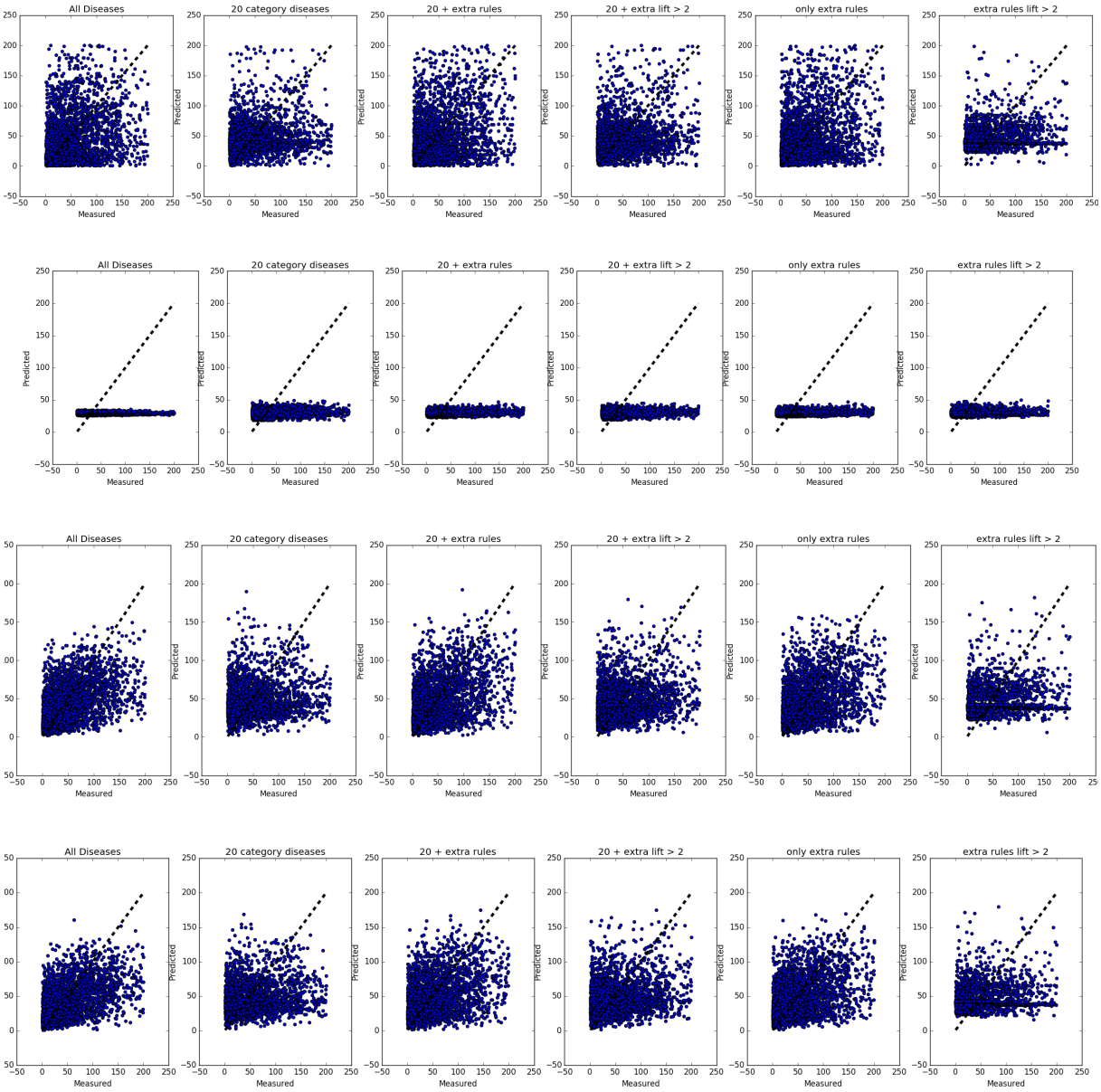


Figura 8.6: Errors de predicció per a tots els models fent servir diferents estratègies.

8.4.3 Freqüències de malalties

Per acabar de comprovar l'anàlisi es va intentar agafar les malalties més rellevants, que apareixien més a la base de dades. A les figures 8.7 i 8.8 es pot veure que el 50% dels diagnòstics a pacients pertany a 18 malalties. La part en verd representa el percentatge d'aparicions d'aquella malaltia i la part blava el total acumulat. Està ordenat de major a menor, essent La depressió i la hipertensió les que apareixen més en aquest conjunt de diagnòstics.

Agafant solament com a variables aquestes malalties no representava una diferència respecte les proves anteriors, concloent que les prediccions obtingudes no poden millorar-se respecte els resultats que proporciona l'aplicació.

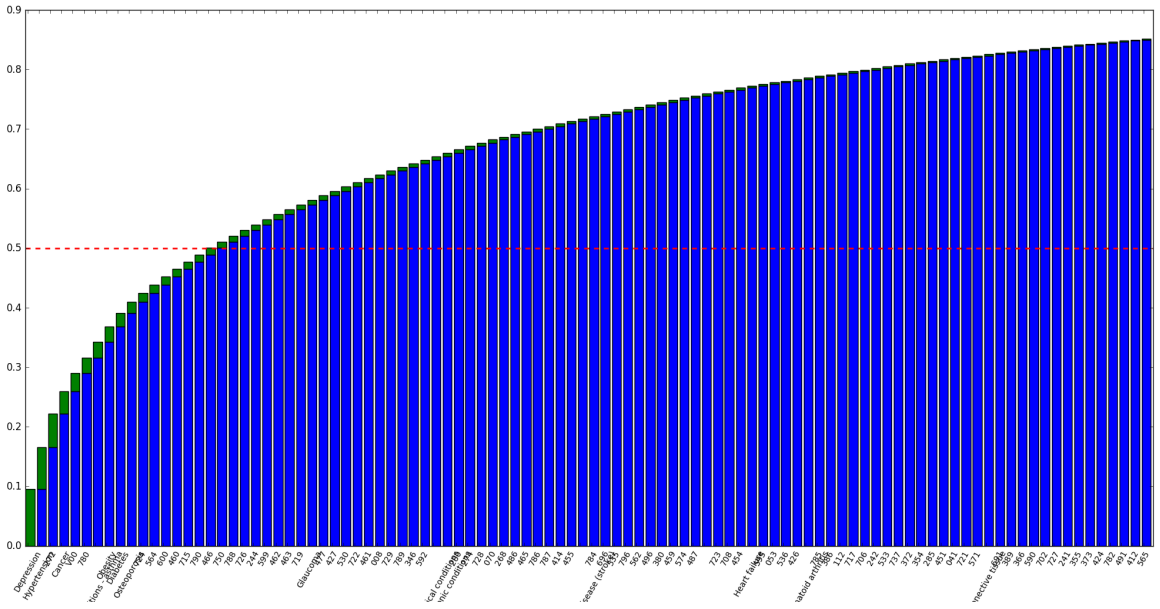


Figura 8.7: Freqüència de malalties de major a menor

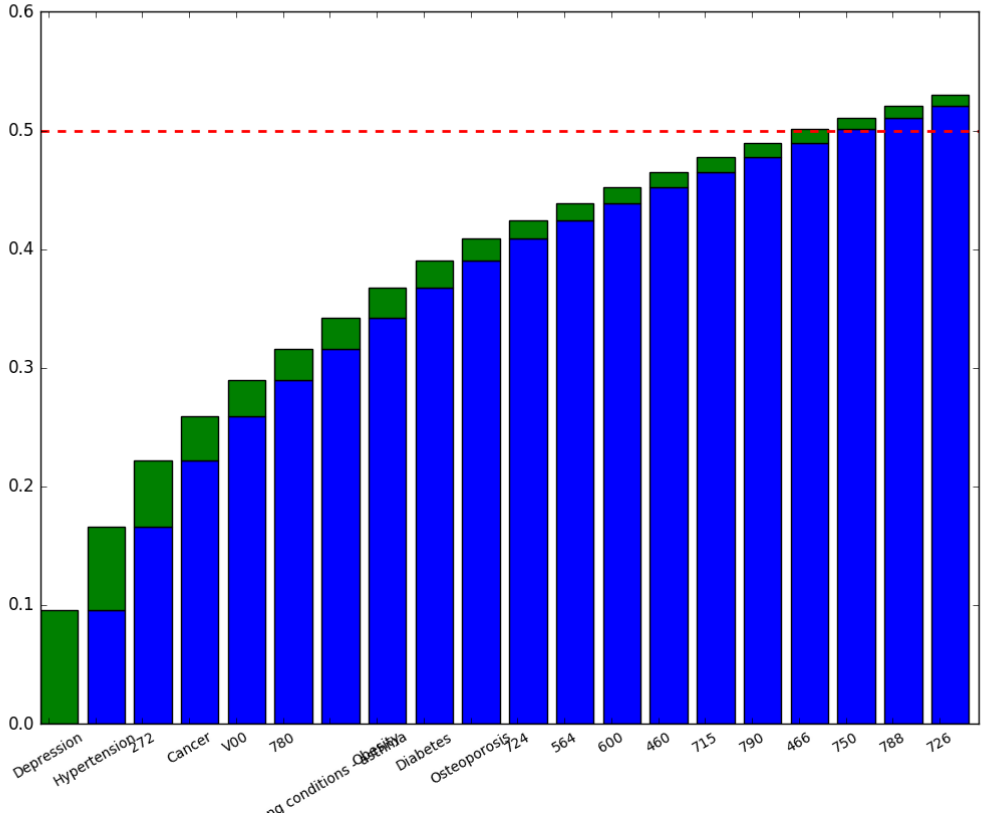


Figura 8.8: 20 malalties més freqüents

8.5 Grafs generats

La generació de grafs de malalties ajuda molt a veure quines són les més freqüents, les que tenen més relacions amb les altres. En total, hi ha 475 malalties diferents, essent les que tenen més connexions amb altres malalties:

Malaltia	**Relacions**
Depressió	394
Hipertensió	355
Transtorns en el metabolisme	349
Càncer	305
Febres, marejos i col·lapses	298
Obesitat	259

Taula 8.5: Relació entre malalties

El diàmetre del graf de malalties és 3, que vol dir que com a màxim hi ha 3 arestes entre 1 node i un altre. Pel que fa a centralitat de nodes, *betweenness* i *closeness*, els que destaquen són els de la taula 8.5, amb petites variacions. Els colors del graf representen comunitats, cada node té un color i està assignat a una comunitat.

A continuació es mostren els grafs generats amb l'aplicació, mostrant primer tot el graf sencer amb totes les malalties i seguit del mateix graf però mostrant els nodes amb més de un cert pes. Aquest pes representa el nombre de pacients que se li han diagnosticat les dues malalties.

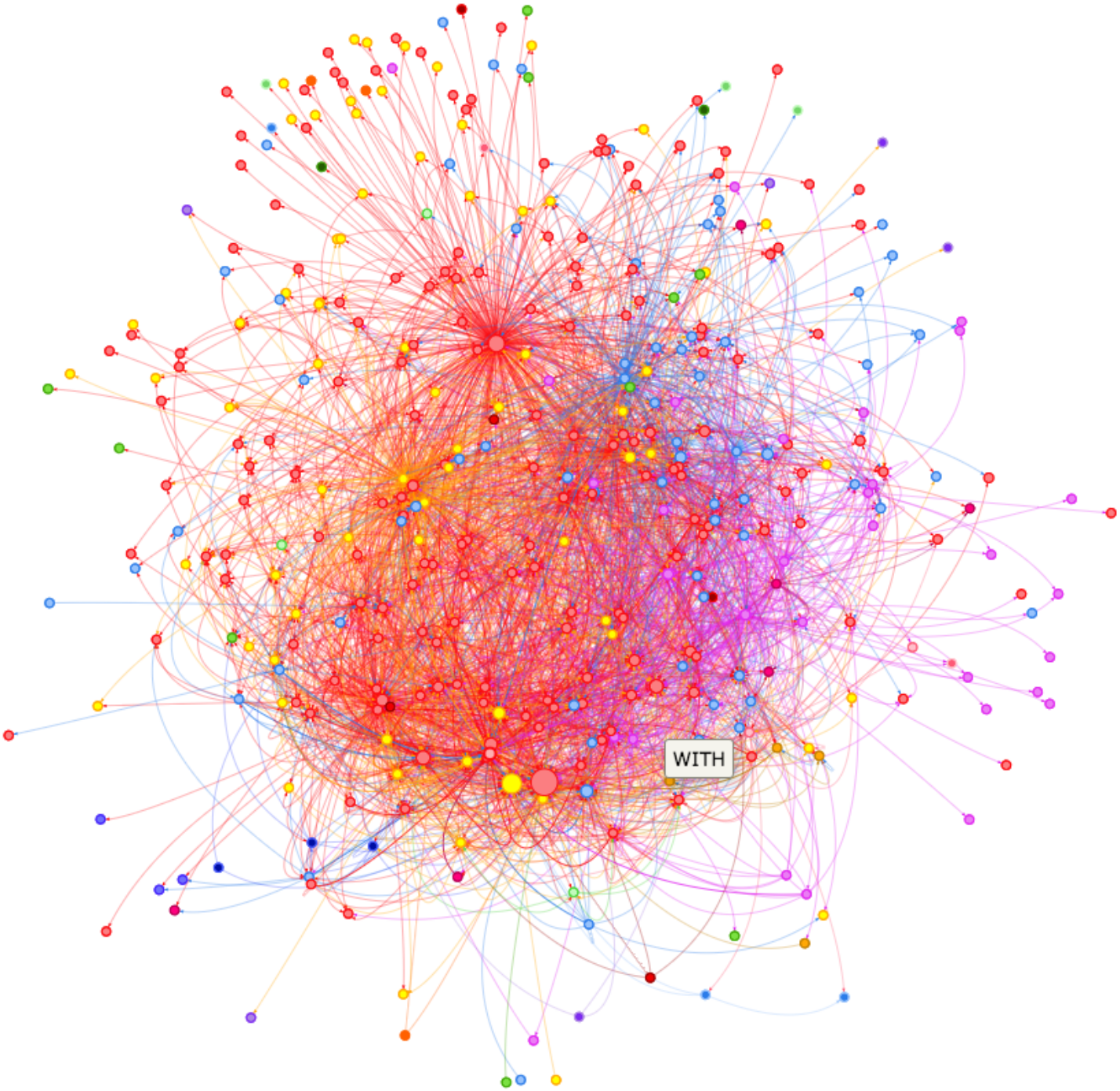


Figura 8.9: Graf de totes les malalties.

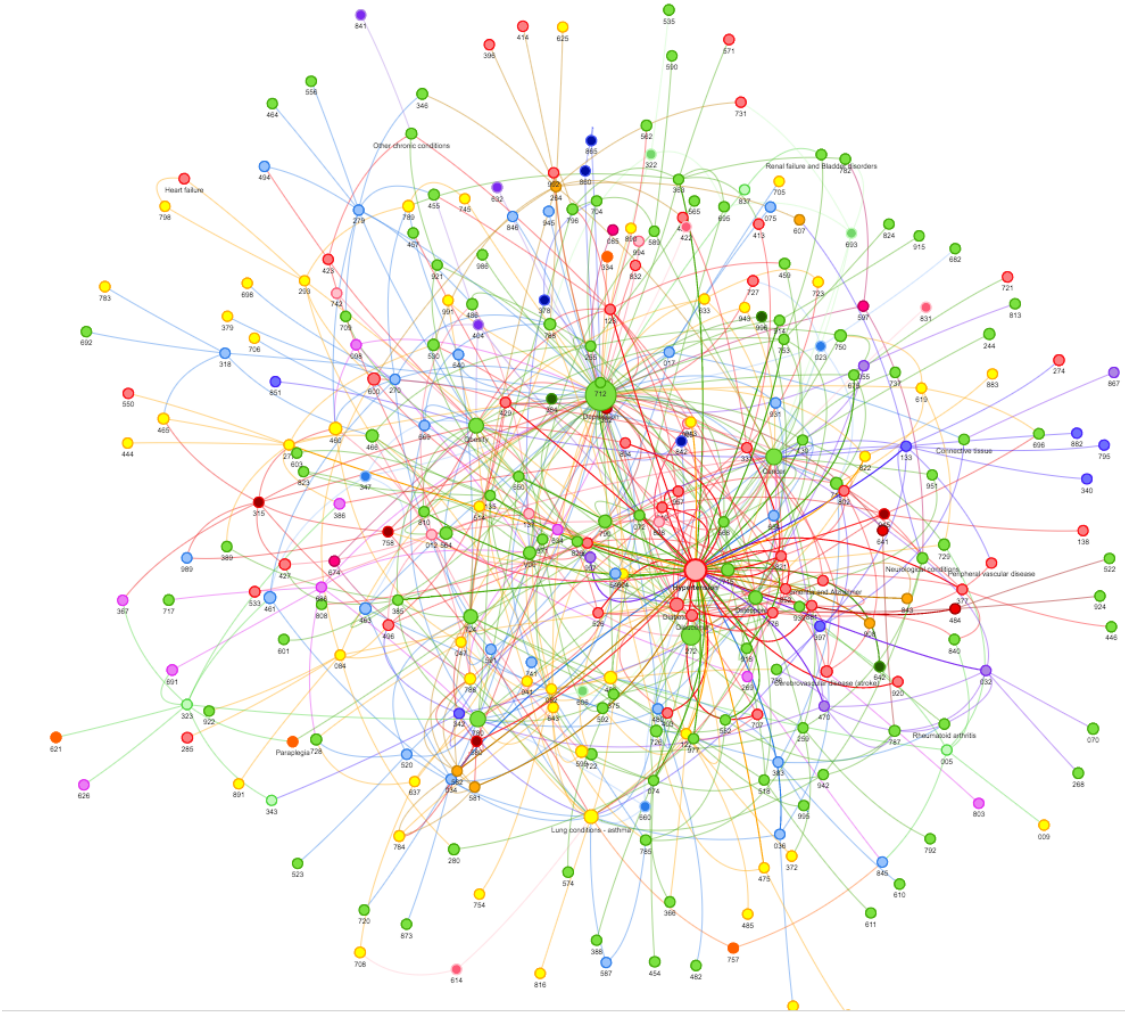
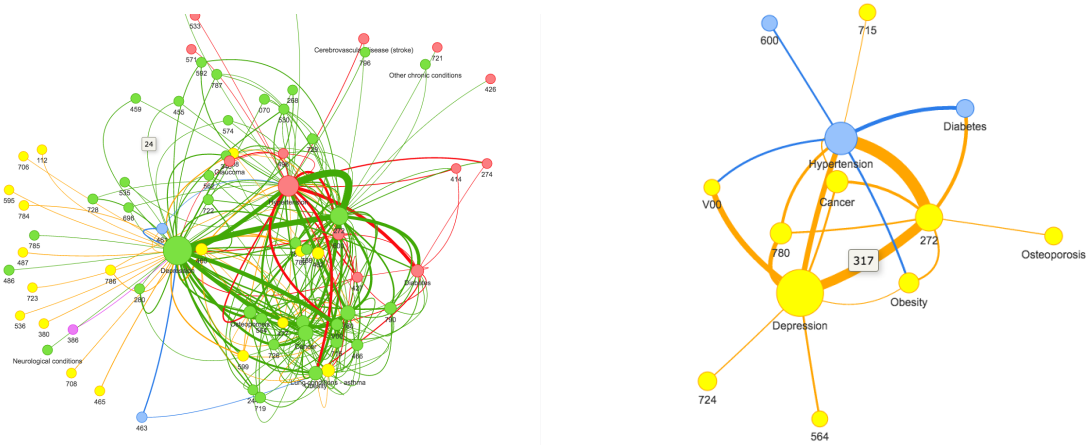


Figura 8.10: Graf amb malalties amb més de 10 arestes.



(a) Graf amb malalties que tenen més de 40 relacions

(b) Graf amb malalties de més de 100 relacions

Figura 8.11: Grafs més detallats

9. Conclusions

L'aplicació s'ha implementat com s'especificava. Personalment creiem que dóna anàlisis interessants pels professionals de la medicina. En aquest sentit, es considera que els objectius s'han assolit.

Els resultats concrets amb les dades proporcionades per Serveis Mèdics han estat menys rics del que es volia, sobretot la predicció i els patrons freqüents. La part del graf ha estat més positiva, s'ha aconseguit una bona representació de les relacions amb les malalties d'aquell conjunt de dades.

9.1 Treball futur

Aquest projecte és un prototip d'aplicació amb les funcionalitats bàsiques per a poder ser d'utilitat. Si es continués, hi ha unes millores que es podrien implementar amb certa facilitat. Tant a nivell intern, com a funcionalitat que oferir o millora de l'experiència de l'usuari. A continuació s'explica un treball que es podria dur a terme.

9.1.1 Seguretat

Actualment l'aplicació es oberta i pot accedir tothom, tant al client com al servidor, base de dades. Es podria fer servir una compta d'usuari per a controlar l'accés a l'aplicació, de manera total o parcial. Podria ser total si l'usuari té accés a tota l'aplicació un cop registrat o bé parcial si es pot donar accés restringit a les funcionalitats, la base de dades o les accions que pot realitzar.

Això seria útil per a tenir l'aplicació a un lloc obert on els usuaris poden registrar-se i fer-la servir però per exemple no es vol que canviïn la base de dades o altra gent no autoritzada la pugui fer servir.

9.1.2 Ampliació dels algorismes utilitzats

Es pot fer fàcilment una ampliació dels algorismes suportats per l'aplicació, ja sigui per a fer patrons freqüents, o bé per a aprenentatge automàtic. El que cal és afegir l'opció d'aquell nou algorisme al client i al servidor implementar un nou mètode que el faci servir amb les dades dels pacients.

9.1.3 Millora de l'experiència de l'usuari

Si s'arribés a seguir el projecte i l'aplicació es provés per usuaris normals, faria falta afegir una gestió d'errors com ara notificacions i alertes. Si no es disposés d'internet o bé el servidor no estigués disponible es necessitaria un avís indican't-ho.

9.1.4 Més flexibilitat en l'entrada de dades

Es podria afegir uns nous camps indicant d'on s'ha d'agafar l'edat, el gènere, el diagnòstic. Ara per ara s'espera que *EDAD* sigui l'edat però es podria especificar d'una altra forma. No seria gaire difícil fer aquesta adaptació i es guanyaria en facilitat d'ús de l'aplicació. Les dades no caldria que es transformessin al format concret desitjat.

9.1.5 Reutilització dels resultats

Els resultats calculats poden tornar a ser útils més endavant i tenir-los guardats és una bona idea. Sobretot si els càlculs per arribar són costosos. Caldria afegir una opció quan es mostren els resultats de guardar permanentment, i una nova vista on es poden seleccionar aquests resultats.

9.1.6 Escalabilitat

S'hauria de veure com funciona l'aplicació amb bases de dades més grans. Si realment els algorismes són eficients i l'interfície es responsiva. El que segurament s'hauria d'implementar a nivell visual és una paginació a l'hora de mostrar els resultats donat que si es fan servir grans quantitats de dades els resultats també seran majors.

9.1.7 Validació de l'aplicació

Es podria provar l'aplicació en usuaris reals, per veure la facilitat d'aprenentatge que suposa i l'usabilitat real que té. Si realment és usable o costa de fer càlculs amb les dades.

Bibliografia

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[BG15] Kaustubh Beedkar and Rainer Gemulla. Lash: Large-scale sequence mining with hierarchies. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 491–503. ACM, 2015.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[CBA15] Karla L Caballero Barajas and Ram Akella. Dynamically modeling patient's health state from electronic medical records: A time series approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 69–78. ACM, 2015.

[CMM+05] Hui Cao, Marianthi Markatou, Genevieve B Melton, Michael F Chiang, and George Hripcsak. Mining a clinical data warehouse to discover disease-finding associations using co-occurrence statistics. In *AMIA*, 2005.

[CNRM14] Yuriy Chechulin, Amir Nazerian, Saad Rais, and Kamil Malikov. Predicting patients with high risk of becoming high-cost healthcare users in ontario (canada). *Healthcare Policy*, 9(3):68, 2014.

[CSPI+14] Jordi Coderch, Inma Sánchez-Pérez, Pere Ibern, Marc Carreras, Xavier Pérez-Berruezo, and José M Inoriza. Predicting individual risk of high healthcare cost to identify complex chronic patients. *Gaceta Sanitaria*, 28(4):292–300, 2014.

[CSPI+16] Marc Carreras, Inma Sánchez-Pérez, Pere Ibern, Jordi Coderch, and José Inoriza. Analysing the costs of integrated care: A case on model selection for chronic care purposes. *International Journal of Integrated Care*, 16(3), 2016.

[GCV+07] Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

[GNPP07] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339. ACM, 2007.

[Gru15] Joel Grus. *Data Science from Scratch: First Principles with Python*. O'Reilly Media, Inc., 1st edition, 2015.

[GSMC10] Akpene Gbegnon, W Nick Street, Jose Monestina, and John W Cromwell. Predicting surgical site infections in real-time. 2010.

[HPYM04] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[HRC09] David A Hanauer, Daniel R Rhodes, and Arul M Chinnaiyan. Exploring clinical associations using '-omics' based enrichment analyses. *PloS one*, 4(4):e5203, 2009.

[JMO+14] Anders Boeck Jensen, Pope L Moseley, Tudor I Oprea, Sabrina Gade Ellesøe, Robert Eriksson, Henriette Schmock, Peter Bjødstrup Jensen, Lars Juhl Jensen, and Søren Brunak. Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. *Nature communications*, 5, 2014.

[LWHX15] Chuanren Liu, Fei Wang, Jianying Hu, and Hui Xiong. Temporal phenotyping from longitudinal electronic health records: A graph based framework. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 705–714, New York, NY, USA, 2015. ACM.

[MFC13] Catia M Machado, Ana T Freitas, and Francisco Couto. Enrichment analysis applied to disease prognosis. *J. Biomedical Semantics*, 4:21, 2013.

[MSL+06] Irene M Mullins, Mir S Siadaty, Jason Lyman, Ken Scully, Carleton T Garrett, W Greg Miller, Rudy Muller, Barry Robson, Chid Apte, Sholom Weiss, et al. Data mining and clinical data repositories: Insights from a 667,000 patient data set. *Computers in biology and medicine*, 36(12):1351–1377, 2006.

[MTIV97] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3):259–289, January 1997.

[PHMA+04] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. on Knowl. and Data Eng.*, 16(11):1424–1440, November 2004.

[RBLA15] Milagros Ruiz, Alex Bottle, Susannah Long, and Paul Aylin. Multimorbidity in hospitalised older patients: Who are the complex elderly? *PloS one*, 10(12):e0145372, 2015.

[RHK10] Naren Ramakrishnan, David Hanauer, and Benjamin Keller. Mining electronic health records. *Computer*, 43(10):77–81, 2010.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag.

[WSW14] Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.

[YL06] Jianji Yang and Judith Logan. A data mining and survey study on diseases associated with paraesophageal hernia. In *AMIA*, 2006.

[Zak00] Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Trans. on Knowl. and Data Eng.*, 12(3):372–390, May 2000.

[Zak01] Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Mach. Learn.*, 42(1-2):31–60, January 2001.

Apèndixs

A. Webs

- **Serveis Mèdics**: http://www.serveismedics.net/cat.
- **Electron**: http://electron.atom.io.
- **Chromium**: https://www.chromium.org.
- **Node.js**: https://nodejs.org.
- **React**: https://facebook.github.io/react.
- **Bulma**: http://bulma.io.
- **Vis.js**: http://visjs.org.
- **Aplicacions fent servir Electron**: http://electron.atom.io/apps.
- **Flask**: http://flask.pocoo.org.
- **Coron system**: http://coron.loria.fr/site.
- **SPMF**:http://www.philippe-fournier-viger.com/spmf.
- **Scikit-learn**: http://scikit-learn.org/stable/index.html.
- **MongoDB**: https://www.mongodb.com.
- **LARCA**: http://recerca.upc.edu/larca.

B. Taula de malalties

A continuació es mostren els codis ICD-9-CM corresponents a les malalties mencionades en aquest projecte. Les 20 malalties amb més impacte no apareixen aquí ja que tenen la seva pròpia taula a la secció 8.1.

Malaltia	**Codis digitals ICD-9-CM**
Transtorns en el metabolisme	272
Taquicàrdies i arritmies	427
Refredat comú, nasofaringitis	460
Bronquitis aguda	466
Transtorns funcionals intestinals i digestius	564
Pròstata	600
Artrosis	715
Lumbàlgia, ciàtica i altres trastorns d'esquena	724
Tendinitis, bursitis i altres entesopaties	726
Hernia hiatal congènita	750
Febres, marejos i col·lapses	780
Incontinència, retenció d'orina, pol·laciúria, poliúria i altres anormalitats de la micció	788
Intolerància a la glucosa, hiperglicèmia i altres nivells anormals en enzims sèrics	790
Sense antecedents, no fumador, sense al·lèrgies conegudes, no consum d'alcohol, sense riscos sanitaris	V00

Taula B.1: Malalties segons els codis ICD-9-CM

C. Instal·lar l'aplicació

Aquesta aplicació per poder-se instal·lar des de zero necessita Node.js, Python v3.5+, Java (per executar les llibreries de patrons freqüents Coron i SPMF) i MongoDB. Els següents enllaços porten a la pàgina d'instal·lació oficial.

- **Node.js**: https://nodejs.org/en/download.
- **Python 3**: https://www.python.org/downloads
- **Java**: http://openjdk.java.net/install
- **MongoDB**: https://docs.mongodb.com/manual/installation.

C.1 Dependències Python

Python necessita tenir instal·lat, els mòduls `Flask`, `scikit-learn`, `numpy`, `scipy`, `python-igraph` i `mongo-connector`. Amb la comanda `pip install` es poden instal·lar sense problemes.

C.2 Preparació de l'entorn

Per a que l'aplicació sigui funcional cal tenir un servidor de MongoDB escoltant el port per defecte que s'assigna (27017), inicialitzar el servidor de Flask amb la comanda `python3 backend/run_server.py` i finalment dins la carpeta de l'aplicació del client (carpeta `client`) fer un `npm install` perquè automàticament s'instal·lin totes les dependències necessàries (pot tardar una estona).

C.3 Execució

Dins la carpeta del client executar la comanda `npm run dev`, s'obrirà una finestra nova amb l'aplicació.